

# A Logic of Interactive Proofs (Formal Theory of Knowledge Transfer)\*

Simon Kramer

University of Luxembourg  
simon.kramer@a3.epfl.ch

June 15, 2012

## Abstract

We propose a logic of interactive proofs as the first and main step towards an intuitionistic foundation for interactive computation to be obtained via an interactive analog of the Gödel-Kolmogorov-Artëmov definition of intuitionistic logic as embedded into a classical modal logic of proofs, and of the Curry-Howard isomorphism between intuitionistic proofs and typed programs. Our interactive proofs effectuate a persistent epistemic *impact* in their intended communities of peer reviewers that consists in the induction of the (propositional) knowledge of their proof goal by means of the (individual) knowledge of the proof with the interpreting reviewer. That is, interactive proofs effectuate a *transfer* of propositional knowledge (knowable facts) via the transmission of certain individual knowledge (knowable proofs) in multi-agent distributed systems. In other words, we as a community can have the formal common knowledge that a proof is that which if known to one of our peer members would induce the knowledge of its proof goal with that member.

**Keywords** cryptographic and interpreted communication; designated-verifier proofs; equality of proofs; interactive and oracle computation; multi-agent distributed systems; normal modal logics; Popper; proofs as sufficient evidence.

## 1 Introduction

The subject matter of this paper is a formal logic of interactive proofs to be used for an intuitionistic foundation of interactive computation.

---

\*Work funded mostly with Grant P 08742 from the Japan Society for the Promotion of Science, and to a lesser extent with Grant AFR 894328 from the National Research Fund Luxembourg cofunded under the Marie-Curie Actions of the European Commission (FP7-COFUND). A preliminary version of Section 1.1 and 1.2 appeared in [Kra10a, not peer-reviewed] and another of Sections 1.1—1.3 in [Kra10b, peer-reviewed].

## 1.1 Motivation, Goal & Problem

### 1.1.1 Motivation

In [GSW06], interactive computation is proposed as the new, *to-be-defined* paradigm of computation, as opposed to the old paradigm of non-interactive computation in the sense of the old sages like Turing and others. The motivation for this paper is the consensus of the contributors to [GSW06], which is that the purpose of interactive computation ultimately is not the computation of result values, to which we consent, but the possibly unending interaction *itself*, from which we dissent. Interaction may well be unending, but it cannot be a self-purpose because if it were then all interactive programs would be quines—rhetorically exaggerated. (A quine program [re]produces itself and only itself.)

### 1.1.2 Goal

Our goal is to reach consensus with the reader that values are only the means—not the ends—of interactive computation, and that **the purpose of interactive computation is *interpreted communication* between distributed man or machine agents interacting via message passing**. Note that a communication channel/medium can be modelled as a machine agent. For example in communication security, which is an important application of interactive computation, the communication medium *is* an adversary.

### 1.1.3 Problem

So what is interpreted communication? According to Shannon [Sha48]:<sup>1</sup>

The fundamental problem of [*uninterpreted*] communication is that of reproducing at one point either exactly or approximately a message selected at another point.

In analogy, we declare:

The fundamental problem of *interpreted* communication is that of [re]producing at one point either exactly or approximately the *intended meaning* of a message selected at another point.

Note that due to the distribution of the different agents in a communication system, which may have different views of the system, the agents constitute different message *interpretation contexts*. Hence, identical messages may well be interpreted differently in different contexts, and thus have different meanings to different agents. As a matter of fact, message misinterpretations are ubiquitous in man or machine communications, e.g., in communication protocols [And08, Chapter 3], and may have serious or even catastrophic consequences,

---

<sup>1</sup>The standard typographic convention of brackets occurring within in-lined or displayed quoted text indicates that the text within the brackets does not occur in the original text. So, “[*uninterpreted*]” indicates that “*uninterpreted*” does not occur in the original text.

e.g., in the context of nuclear command and control [And08, Chapter 13]. Indeed, [re]producing intended message meaning across interpretation contexts is a highly critical and non-trivial problem. But what does message meaning mean more precisely?

In [Kra07b], we argue that the (denotational) meaning of a message in a given interpretation context is the *propositional knowledge* that the *individual knowledge* of that message induces in that context (cf. Page 21 and Section 2.1 here). (See [PR03] for a related notion of message meaning.) By individual knowledge we mean knowledge in the sense of the transitive use of the verb “to know”, here to know a message, such as the plaintext of an encrypted message. Notation:  $a \text{ } k \text{ } M$  for “agent  $a$  knows message  $M$ ” (cf. Definition 1). This is the classic concept of knowledge *de re* (“of a thing”) made explicit for message things. Whereas by propositional knowledge we mean knowledge in the sense of the use of the verb “to know” with a clause, here to know that a statement is true, such as that the plaintext of an encrypted message is (individually) unknown to potential adversaries. Notation:  $K_a(\phi)$  for “agent  $a$  knows that  $\phi$  [is true]” (cf. Section 2.1). This is the classic concept of knowledge *de dicto* (“of a fact”).<sup>2</sup> Notice that we make the distinction between individual and propositional knowledge with respect to the “*object*” of knowledge (the *known*), i.e., with respect to a message and clause, respectively. However, individual as well as propositional knowledge can both be individual with respect to the *subject* of knowledge (the *knower*), i.e., an (individual) agent.

Hence, an agent-centric paraphrase of our previous problem statement is:

The fundamental problem of communication is that of inducing at one point either an intended knowledge or an intended belief with a message selected at another point (cf. Section 2.1 for formal meanings).

With this paper, we intend to induce (necessarily true) knowledge, and leave induction of (possibly false) belief for further work. (For our standard notions of belief and knowledge, see [MV07].) Here, interactive computations compute propositional *knowledge* (e.g., that the goal of this paper has been achieved), and they do so by passing as messages pieces of interactively or non-interactively computed individual knowledge (e.g., this paper). Again, result values are only the means—not the ends—of interactive computations.

## 1.2 Solution & Methodology

### 1.2.1 Solution

Our problem statement contains an inceptive solution and defining principle for interactive computation, namely *induction of knowledge* (cf. Section 2.1). Our task is thus to make this principle precise. This in turn leads us to defining the concept of an *interactive proof* (or *certificate*) whose effect is to induce the knowledge of its proof goal (or statement of certification) in the intended

---

<sup>2</sup> In a first-order setting, knowledge *de re* and *de dicto* can be related in Barcan-laws.

interpretation context (cf. Section 2.1). The present paper is intended to be such an interactive proof: its proof goal is the goal stated in Section 1.1.2, and its intended interpretation context is the set of logically educated readers fluent in English. Our interactive proofs are also *formal social* proofs in that they partially reconcile two distinct viewpoints on mathematical proofs [Bus98a]:

The first view is that proofs are social conventions by which mathematicians convince one another of the truth of theorems. That is to say, a proof is expressed in natural language plus possibly symbols and figures, and is sufficient to convince an expert of the correctness of a theorem. Examples of social proofs include the kinds of proofs that are presented in conversations or published in articles. Of course, it is impossible to precisely define what constitutes a valid proof in this social sense; and, the standards for valid proofs may vary with the audience and over time. The second view of proofs is more narrow in scope: in this view, a proof consists of a string of symbols which satisfy some precisely stated set of rules and which prove a theorem, which itself must also be expressed as a string of symbols. According to this view, mathematics can be regarded as a ‘game’ played with strings of symbols according to some precisely defined rules. Proofs of the latter kind are called “formal” proofs to distinguish them from “social” proofs.

Note that a theorem known by one (say  $a$ ) but not by another mathematician (say  $b$ ) is a local truth from the viewpoint of an audience (say  $\{a, b\}$ ). An example of a social convention is a work contract (cf. Lemma 3 and Corollary 5).

### 1.2.2 Methodology

Our methodology for defining interactive computation emerges as an interactive variant of a classical construction that consists in a “horizontal” transitive *embedding* of programs into proofs and in a “vertical” homomorphism of each non-interactive structure into its interactive counterpart (cf. Figure 1). We will argue that the right-most “vertical” homomorphism (without  $\subset$ -tail) cannot be an embedding (with  $\subset$ -tail) and that this reflects the essential difference between interactivity and non-interactivity here. More precisely, we shall present:

1. **in this paper**, a classical modal logic (LiP) of *interactive proofs* that
  - (a) are agent-centric generalisations of non-interactive proofs such that the agents are resource-*unbounded* with respect to individual and thus also propositional knowledge (cf. Section 3.2.2), though our agents here are still unable to guess individual (and thus also propositional) knowledge
  - (b) induce the knowledge of their proof goal with their intended interpreting agent(s) such that the induced knowledge is propositional in the sense of the standard modal logic of knowledge S5 [FHMV95, HR10]

Figure 1: Typed interactive programs from interactive proofs



(See [Kra07a, Kra08a] and [Kra08b] for preliminary, non-axiomatic explorations within different, non-standard semantics, but in [Kra08b] already with pairing and signing as proof-term constructors.)

## 2. in future work:

- (a) a classical modal logic (iS4) of *interactive provability* via an embedding into LiP in analogy with Artëmov's embedding of the standard modal logic of non-arithmetic<sup>3</sup> provability S4 into his Logic of Proofs LP [Art94, Art01, Art07]
- (b) *interactive Intuitionistic Logic* (iIL) via an embedding into iS4 in analogy with the Gödel-Kolmogorov embedding of Intuitionistic Logic IL into S4 [Art07]
- (c) *typed interactive programs* (tiP) via an isomorphism from iIL in analogy with the Curry-Howard isomorphism between IL and typed programs tP [dG95].

We will deploy our methodology from right to left. LiP (LP) is the richest among all the (non-)interactive structures in the sense that all other (non-)interactive structures embed into LiP (LP). In result, terms viewed as proofs are descriptions of constructive deductions, terms viewed as programs are prescriptions for interactive computations, LiP-formulas viewed as propositions are proof goals, and LiP-formulas viewed as types are program properties. To agents, interactive proofs are message terms that induce the propositional knowledge of their proof goal with their intended interpreters, and interactive computations are message communications between distributed interlocutors that compute that knowledge from the meaning of the communicated messages. In sum, *the purpose of interactive proofs is the transfer of propositional knowledge (knowable facts) via the transmission of certain individual knowledge (knowable proofs) in multi-agent distributed systems* (e.g., editorial boards, scientific communities, social networks and other virtualised societies—even the whole Internet). That is, ***LiP is a formal theory of knowledge transfer***. In contrast, Shannon's theory is about the (error-correcting) transmission of individual knowledge (i.e., data) only.

<sup>3</sup>i.e., not internalising provability of a formal system that includes Peano Arithmetic

### 1.3 The Logic of Proofs (LP)

The language of Artëmov's Logic of Proofs (LP) (cf. [Art94, Art01] and [Art07, Section 5]) is the language of classical propositional logic enriched with formulas  $p:F$ , where  $F$  denotes formulas and  $p$  so-called proof polynomials. Proof polynomials are terms built from proof variables  $x, y, z, \dots$  and proof constants  $a, b, c, \dots$  by means of three operations: application  $\cdot$  (binary), sum  $+$  (binary), and proof checker  $!$  (unary). According to Artëmov, proof polynomials represent the whole set of possible operations on (non-interactive) proofs for a propositional language.

Then, the following proof system defines (the non-normal modal logic) LP:

0. all axioms of classical propositional logic
1.  $\vdash_{\text{LP}} ((p:F) \vee q:F) \rightarrow (p+q):F$  (sum)
2.  $\vdash_{\text{LP}} (p:(F \rightarrow G)) \rightarrow ((q:F) \rightarrow (p \cdot q):G)$  (application)
3.  $\vdash_{\text{LP}} (p:F) \rightarrow F$  (reflection)
4.  $\vdash_{\text{LP}} (p:F) \rightarrow (!p):(p:F)$  (proof checker)
5.  $\{F \rightarrow G, F\} \vdash_{\text{LP}} G$  (*modus ponens*)
6.  $\vdash_{\text{LP}} c:A$ , for any axiom  $A$  and proof constant  $c$  (constant specification<sup>4</sup>),

where  $\{F \rightarrow G, F\} \vdash_{\text{LP}} G$  abbreviates “if  $\vdash_{\text{LP}} F \rightarrow G$  and  $\vdash_{\text{LP}} F$  then  $\vdash_{\text{LP}} G$ ” in horizontal Hilbert style, and  $!$  is interpreted as a primitive-recursive program for checking the correctness of proofs which given a proof of  $p$  produces a proof that  $p$  proves  $F$ . The application axiom internalises the *modus ponens* rule.

Note that LP does not explicate beyond the formula  $p:F$  what it means for  $p$  to prove  $F$ , but rather attempts to characterise axiomatically this relation. Indeed,  $p:F$  really stands for an atomic concept. Arguably, the standard semantics of LP does not fully explicate the concept either: that semantics actually merely re-stipulates each axiom of LP as a corresponding condition on the model in set-theoretic language (cf. [Art07, Section 5.3] and [Fit05]). In that, it is rather a convenient semantic *interface*—and we will use it as such—than a semantics proper. Here it is. Given

- a frame  $(W, R)$  with a reflexive and transitive relation  $R \subseteq W \times W$
- an abstractly constrained *evidence mapping*  $\mathcal{E}$  from worlds  $u$  and proof polynomials  $p$  to sets of formulas  $F$

such that:

1. if  $uRv$  then  $\mathcal{E}(u, p) \subseteq \mathcal{E}(v, p)$  (monotonicity)
2. (closure)

---

<sup>4</sup>Constant specification is a somewhat flexible concept (cf. [Art08a] for four variations).

- if  $F \rightarrow G \in \mathcal{E}(u, p)$  and  $F \in \mathcal{E}(u, q)$  then  $G \in \mathcal{E}(u, p \cdot q)$  (application)
- if  $F \in \mathcal{E}(u, p)$  then  $p:F \in \mathcal{E}(u, !p)$  (proof checker)
- $\mathcal{E}(u, p) \cup \mathcal{E}(u, q) \subseteq \mathcal{E}(u, p+q)$  (sum),

and a usual valuation mapping  $\mathcal{V}$  from atomic propositions to sets of worlds, satisfaction for the LP-modality in a model  $(W, R, \mathcal{E}, \mathcal{V})$  at a world  $u$  is so that

$$(W, R, \mathcal{E}, \mathcal{V}), u \Vdash p:F \text{ iff} \\ \text{for every } v \in W, \text{ if } uRv \text{ and } F \in \mathcal{E}(u, p) \text{ then } (W, R, \mathcal{E}, \mathcal{V}), v \Vdash F.$$

Notice the additional constraint  $F \in \mathcal{E}(u, p)$ , which a standard Kripke-semantics format would not allow. A more serious criticism than the one of not being a semantics proper is that in a truly interactive setting, the reflection axiom is unsound (cf. Section 3.1). By a truly interactive setting we mean a multi-agent distributed system where not all proofs are known by all agents, i.e., a setting with a non-trivial distribution of information (in the sense of Dana Scott, cf. Proposition 3).

In contrast:

1. LiP will give an *epistemic explication* of proofs, i.e., an explication of proofs in terms of the epistemic impact that they effectuate with their intended interpreting agents (i.e., the knowledge of their proof goal).

Technically, we will endow the proof modality with a *standard* Kripke-semantics, whose accessibility relation we

- (a) define *constructively*, in terms of elementary set-theoretic constructions (in loose analogy with the constructive rather than the purely axiomatic definition of ordered pairs [e.g., Kuratowski's] or numbers [Fef89])
- (b) match to a simplified and then interactively generalised version of the semantic interface of LP, where the simplification consists of
  - i. the absorption of the evidence mapping into the accessibility relation (and thus the absorption of the corresponding conjunctive constraint on the truth condition of the proof modality)
  - ii. the elimination of the monotonicity constraint on the evidence mapping (in the sense that the constraint will become a property), which is a nice side-effect of the previous simplification

2. LiP only validates a corresponding *conditional reflection principle*, i.e., a reflection principle that is conditioned on the (individual) knowledge of the proof mentioned by the principle (e.g., the above  $p$  in LP).
3. LiP is, technically speaking, a *normal* modal logic, which brings all the benefits of the existing standard techniques of normal modal logics to LiP.

Hence, we beg to differ with Artëmov and Nogina, who, like Aristotle and Plato, define (propositional) knowledge as justified true belief, but unlike Aristotle and Plato, admit as admissible justifications for such knowledge only proofs in the sense of at least LP [AN05, Art08a]. As a counter-example to *Artëmov and Nogina’s provability explication of knowledge*, consider that an agent may know that a certain state of affairs is the case from the *observation of a physical event* (e.g., a message input/output), yet not be able to prove her (propositional) knowledge to the non-observers (e.g., an absent peer or judge) for lack of *sufficient evidence* (i.e., proof). Whereas in *our epistemic explication of provability*, provability possibly implies propositional knowledge, e.g., with the (individual) knowledge of a proof, but propositional knowledge does not necessarily imply provability, e.g., without such a proof. The technical difference between the two philosophies may be subtle but nevertheless is serious (i.e., not a mere technicality)—especially for applications to truly distributed computer systems.

## 1.4 Contribution & Roadmap

### 1.4.1 Contribution

The contribution of this paper is a *formal theory of knowledge transfer*, i.e., the classical normal modal *Logic of interactive Proofs (LiP)*, to be used to define the new paradigm of interactive computation via a classic construction due to Gödel-Kolmogorov-Artëmov. More precisely, our main contributions are:

1. a *constructive Kripke-semantics* for LiP’s proof modality (cf. Page 19)
2. a *sound and complete axiomatisation* for LiP (cf. Theorem 5)
3. a stateful notion of transmittable *interactive proofs* that
  - (a) are *agent-centric generalisations* of non-interactive proofs such that the agents are, as said, still *resource-unbounded* with respect to individual and thus also propositional knowledge
  - (b) have intuitive *epistemic explications* in that
    - i. they effectuate (cf. Section 2.1)
      - A. a persistent epistemic *impact* in their intended communities of peer reviewers that consists in the induction of the (propositional) knowledge of their proof goal by means of the (individual) knowledge of the proof with the interpreting reviewer
      - B. a *transfer* of propositional knowledge (knowable facts) via the transmission of certain individual knowledge (knowable proofs) in multi-agent distributed systems
    - ii. the individual proof knowledge can be thought of as being provided by an imaginary *computation oracle* (cf. Section 2.2)
  - (c) are *falsifiable* in a communal sense of Popper’s (cf. Theorem 4)



- (d) can be constructed with only two operations, namely *pairing* and *signing*, and freely combined with other term operations (e.g., *encryption*)
  - (e) happen to have an *information-theoretic explication* in terms of Scott’s information systems (cf. Proposition 3)
4. a stateful notion of *proof equality* in an idempotent commutative monoid capturing ***equality of epistemic impact*** (cf. Corollary 4)
  5. a *novel modal rule of logical modularity*, called *epistemic antitonicity*, for the class of justification logics [Art08a] including LP, which allows the partial, or even total and thus modular generation of the structural modal laws from the laws of a separate (e.g., application-specific) term theory (cf. Page 11 and Section 3.2.1).

In sum, LiP is a *minimal modular extension* of propositional logic with

1. an interactively generalised additional operator (the proof modality)
2. a simplified and then interactively generalised
  - (a) proof-term language (only two instead of three constructors, ***agents as proof- as well as signature-checkers***)
  - (b) constructive Kripke-semantics (including evidence-mapping absorption and monotonicity-condition elimination).

With our contribution, we mean to concur with [Mos06, Page viii], where

computation theory is viewed as part of the mathematics “to be founded,”

since Kripke-models such as ours for LiP—conceived as a foundation for *interactive* computation theory—are relational models of the meaning of modal languages in the language of set theory, which in turn [Mos06, Page vii]

is the official language of mathematics, just as mathematics is the official language of science.

### 1.4.2 Roadmap

In the next section, we introduce our Logic of interactive Proofs (LiP) axiomatically by means of a compact closure operator that induces the Hilbert-style proof system that we seek (cf. Proposition 1) and that allows the simple generation of application-specific extensions of LiP (cf. Page 12). We then prove some useful (further-used), deducible laws within the obtained system. Next, we introduce the constructive semantics and the semantic interface for LiP. For the construction of the semantics, we again make use of a closure operator, but this time on sets of messages to be used as interactive proofs. In Section 2.1, we present the promised epistemic explication and in Section 2.2 the promised

oracle-computational explication of our interactive proofs. In Section 2.3, we demonstrate the adequacy of our proof system and present our notion of proof equality for LiP. Finally, we relate LiP to LP-like systems in Section 3.

## 2 Basic Logic of interactive Proofs

The basic Logic of interactive Proofs (LiP) provides a modal *formula language* over a generic message *term language*. The formula language offers the propositional constructors, a relational symbol ‘ $k$ ’ for constructing atomic propositions about individual knowledge, and a parameterised unary modal constructor ‘ $\cdot$ ’ for propositions about proofs. The message language offers term constructors for message *pairing* and (not necessarily, but possibly cryptographically implemented) *signing*. (Cryptographic signature creation and verification is polynomial-time computable [Kat10].)

**Definition 1** (The language of LiP). Let

- $\mathcal{A} \neq \emptyset$  designate a non-empty finite set of *agent names*  $a, b, c$ , etc.
- $\mathcal{C} \subseteq \mathcal{A}$  denote (finite and not necessarily disjoint) communities of agents  $a \in \mathcal{A}$  (referred to by their name)
- $\mathcal{M} \ni M ::= a \mid B \mid \llbracket M \rrbracket_a \mid (M, M)$  designate our language of *message terms*  $M$  over  $\mathcal{A}$  with (transmittable) agent names  $a \in \mathcal{A}$ , application-specific data  $B$  (left blank here), signed messages  $\llbracket M \rrbracket_a$ , and message pairs  $(M, M)$   
(Messages must be grammatically well-formed, which yields an induction principle. So agent names  $a$  are logical term constants, the meta-variable  $B$  just signals the possibility of an extended term language  $\mathcal{M}$ ,  $\llbracket \cdot \rrbracket_a$  with  $a \in \mathcal{A}$  is a unary functional symbol, and  $(\cdot, \cdot)$  a binary functional symbol.)
- $\mathcal{P}$  designate a denumerable set of *propositional variables*  $P$  constrained such that for all  $a \in \mathcal{A}$  and  $M \in \mathcal{M}$ ,  $(a k M) \in \mathcal{P}$  (for “ $a$  knows  $M$ ”) is a distinguished variable, i.e., an *atomic proposition*, (for *individual* knowledge)  
(So  $a k \cdot$  where  $a \in \mathcal{A}$  is a unary relational symbol.)
- $\mathcal{L} \ni \phi ::= P \mid \neg \phi \mid \phi \wedge \phi \mid M :_a^{\mathcal{C}} \phi$  designate our language of *logical formulas*  $\phi$ , where  $M :_a^{\mathcal{C}} \phi$  means that “ $M$  is a  $\mathcal{C} \cup \{a\}$ -reviewable proof of  $\phi$ ” in the sense that “ $M$  can prove  $\phi$  to  $a$  (e.g., a designated verifying judge) and this fact is commonly known in the (pointed) community  $\mathcal{C} \cup \{a\}$  (e.g., for  $\mathcal{C}$  being a jury).”

LiP is defined by means of the following axiom and deduction-rule schemas.

**Definition 2** (The axioms and deduction rules of LiP). Let

- $\Gamma_0$  designate an adequate set of axioms for classical propositional logic

- $\Gamma_1 := \Gamma_0 \cup \{$ 
  - $a \text{ k } a$  (knowledge of one’s own name string)
  - $a \text{ k } M \rightarrow a \text{ k } \llbracket M \rrbracket_a$  (*personal* [the same  $a$ ] signature *synthesis*)
  - $a \text{ k } \llbracket M \rrbracket_b \rightarrow a \text{ k } (M, b)$  (*universal* [any  $a$  and  $b$ ] signature *analysis*)
  - $(a \text{ k } M \wedge a \text{ k } M') \leftrightarrow a \text{ k } (M, M')$  ([un]pairing)
  - $(M :_a^C (\phi \rightarrow \phi')) \rightarrow ((M' :_a^C \phi) \rightarrow (M, M') :_a^C \phi)$  (GK)<sup>5</sup>
  - $(M :_a^C \phi) \rightarrow (a \text{ k } M \rightarrow \phi)$  (epistemic truthfulness)
  - $(M :_a^C \phi) \rightarrow \bigwedge_{b \in \mathcal{C} \cup \{a\}} (\llbracket M \rrbracket_a :_b^{C \cup \{a\}} (a \text{ k } M \wedge M :_a^C \phi))$  (peer review)
  - $(M :_a^{C \cup \mathcal{C}'} \phi) \rightarrow M :_a^C \phi$  (group decomposition)  $\}$

designate a set of *axiom schemas*.

Then,  $\text{LiP} := \text{Cl}(\emptyset) := \bigcup_{n \in \mathbb{N}} \text{Cl}^n(\emptyset)$ , where for all  $\Gamma \subseteq \mathcal{L}$ :

$$\begin{aligned}
\text{Cl}^0(\Gamma) &:= \Gamma_1 \cup \Gamma \\
\text{Cl}^{n+1}(\Gamma) &:= \text{Cl}^n(\Gamma) \cup \\
&\quad \{ \phi' \mid \{ \phi, \phi \rightarrow \phi' \} \subseteq \text{Cl}^n(\Gamma) \} \cup \quad (\textit{modus ponens}) \\
&\quad \{ M :_a^C \phi \mid \phi \in \text{Cl}^n(\Gamma) \} \cup \quad (\textit{necessitation}) \\
&\quad \{ (M' :_a^C \phi) \rightarrow M :_a^C \phi \mid (a \text{ k } M \rightarrow a \text{ k } M') \in \text{Cl}^n(\Gamma) \} \\
&\quad (\textit{epistemic antitonicity}).
\end{aligned}$$

We call LiP the *base theory*, and  $\text{Cl}(\Gamma)$  an *LiP-theory* for any  $\Gamma \subseteq \mathcal{L}$ .

This article is about the base theory (the logic), as suggested by the article title. Notice the logical order of LiP, which is, due to propositions about (proofs of) propositions, *higher-order propositional*. Further, observe that we assume the existence of a dependable mechanism for signing messages, which we model with the above synthesis and analysis axioms. In *trusted* multi-agent distributed systems, signatures are *unforged*, and thus such a mechanism is trivially given by the inclusion of the sender’s name in the sent message, or by the sender’s sensorial impression on the receiver when communication is immediate. In *distrusted* multi-agent distributed systems (e.g., the open Internet), a practically *unforgeable* signature mechanism can be implemented with classical *certificate-based* or, more directly, with *identity-based* public-key cryptography [Kat10]. We also assume the existence of a pairing mechanism modelling finite sets. Such a mechanism is required by the important application of communication (not only cryptographic) protocols [And08, Chapter 3], in which concatenation of high-level data packets is associative, commutative, and idempotent. As examples of application-specific data  $B$  we conceive of:

<sup>5</sup>“GK” abbreviates “Generalised Kripke-law”.

Table 1: Some macro-definable proof concepts

$M \div_a^{\mathcal{C}} \phi := M :_a^{\mathcal{C}} \neg \phi$
$(M \text{ is a } \mathcal{C} \cup \{a\}\text{-reviewable } \textit{refutation} \text{ of } \phi \text{ to } a)$
$M \diamond_a^{\mathcal{C}} \phi := \neg(M \div_a^{\mathcal{C}} \phi)$
$(M \text{ is a } \mathcal{C} \cup \{a\}\text{-reviewable } \textit{proof diamond} \text{ of } \phi \text{ to } a)$
$M \vee_a^{\mathcal{C}} \phi := (M :_a^{\mathcal{C}} \phi) \vee (M \div_a^{\mathcal{C}} \phi)$
$(M \text{ is a } \mathcal{C} \cup \{a\}\text{-reviewable } \textit{decider} \text{ of } \phi \text{ to } a)$
$M \bar{\wedge}_a^{\mathcal{C}} \phi := \neg(M \vee_a^{\mathcal{C}} \phi)$
$(M \text{ is a } \mathcal{C} \cup \{a\}\text{-reviewable } \textit{non-decider} \text{ of } \phi \text{ to } a)$

- *atomic data* other than agent names such as random numbers (systematically used in cryptographic communication), quoted formulas  $\ulcorner \phi \urcorner$  (e.g., the Gödel-number of  $\phi$  in some Gödel-numbering scheme)<sup>6</sup>, and others;
- *compound data* such as

- hashed<sup>7</sup> data  $\lceil M \rceil$ , for  $M \in \mathcal{M}$  and with axiom  $a \mathbf{k} M \rightarrow a \mathbf{k} \lceil M \rceil$
- encrypted data  $\lceil M \rceil_{M'}$ , for plaintext data  $M \in \mathcal{M}$  and data used as an encryption key  $M' \in \mathcal{M}$ , and with axioms
  - \*  $a \mathbf{k} (M, M') \rightarrow a \mathbf{k} \lceil M \rceil_{M'}$  (encryption)
  - \*  $a \mathbf{k} (\lceil M \rceil_{M'}, M') \rightarrow a \mathbf{k} M$  (decryption)

This is the so-called Dolev-Yao conception of cryptography [DY83], which we could easily cast as the following LiP-theory<sup>8</sup>

$$\text{LiP}_{\text{DY}} := \text{Cl}(\{a \mathbf{k} M \rightarrow a \mathbf{k} \lceil M \rceil, \\ a \mathbf{k} (M, M') \rightarrow a \mathbf{k} \lceil M \rceil_{M'}, \\ a \mathbf{k} (\lceil M \rceil_{M'}, M') \rightarrow a \mathbf{k} M\}).$$

Now note the following macro-definitions:  $\top := a \mathbf{k} a$ ,  $\perp := \neg \top$ ,  $\phi \vee \phi' := \neg(\neg \phi \wedge \neg \phi')$ ,  $\phi \rightarrow \phi' := \neg \phi \vee \phi'$ ,  $\phi \leftrightarrow \phi' := (\phi \rightarrow \phi') \wedge (\phi' \rightarrow \phi)$ , and, more interestingly those in Table 1.<sup>9</sup> Variations on our notions of interactive proof can also be macro-defined, e.g., with respect to *reviewer communities* (by conjunction with respect to their members and based on a policy of either one

<sup>6</sup>Quotation is a form of type down-casting in the sense that data viewed as compound at a certain logical level (here, at the formula-language level) is viewed as atomic at a lower level (here, at the term-language level), and thus is a form of encoding meta-data (here, statements about messages) in object data (here, messages).

<sup>7</sup>Cryptographic hash functions are one-way functions with certain cryptographically interesting properties such as collision and pre-image resistance.

<sup>8</sup>The integration of other conceptions such as the classical information-theoretic [Sha49] and the modern complexity-theoretic [Gol01, Gol04] will be presented in future work.

<sup>9</sup>The problem of defining interactive refutations was suggested to me by Rajeev Goré.

[dis]proof for *all* members or one [dis]proof for *each* member) and with respect to *exclusive communities* (with respect to members only).

We could also conceive of *term forms*, i.e., terms containing *free variables* (data place holders)  $x, x', x'' \in \mathcal{X}$  in appropriate places, where  $\mathcal{X}$  would designate a countably infinite set of variables. However, in order to keep the introduction of our logic as simple as possible and as complicated as necessary, we do not officially introduce term forms here but content ourselves with stating the following reasonable axiom schemas for the treatment of free variables:

- $a \mathbf{k} M[M'/x] \rightarrow a \mathbf{k} M(x)$  (unplugging)
- $a \mathbf{k} (M(x), M') \rightarrow a \mathbf{k} M[M'/x]$  (plugging),

where  $M[M'/x]$  designates the simultaneous substitution of the term form  $M'$  for all free occurrences of the term variable  $x$  in the term form  $M$ . Of course, one could introduce instead term-variable binders ( $\lambda$ -abstractors) [HS08] and work with bound variables, in order to make the function character of term forms explicit.

Finally, we could close individual knowledge under an *equational theory* defined by atomic propositions  $(M = M') \in \mathcal{P}$ , by adding the axiom schema

$$(a \mathbf{k} M \wedge M = M') \rightarrow a \mathbf{k} M' \quad (\text{equational closure}).$$

Note that in the sequel, “:iff” abbreviates “by definition, if and only if”. Logicians may want to skip the following proposition.

**Proposition 1** (Hilbert-style proof system). *Let*

$$\begin{aligned} \Phi \vdash_{\text{LiP}} \phi & \text{ :iff } \text{ if } \Phi \subseteq \text{LiP} \text{ then } \phi \in \text{LiP} \\ \phi \dashv\vdash_{\text{LiP}} \phi' & \text{ :iff } \{\phi\} \vdash_{\text{LiP}} \phi' \text{ and } \{\phi'\} \vdash_{\text{LiP}} \phi \\ \vdash_{\text{LiP}} \phi & \text{ :iff } \emptyset \vdash_{\text{LiP}} \phi. \end{aligned}$$

In other words,  $\vdash_{\text{LiP}} \subseteq 2^{\mathcal{L}} \times \mathcal{L}$  is a system of closure conditions in the sense of [Tay99, Definition 3.7.4]. For example:

1. for all axioms  $\phi \in \Gamma_1$ ,  $\vdash_{\text{LiP}} \phi$
2. for modus ponens,  $\{\phi, \phi \rightarrow \phi'\} \vdash_{\text{LiP}} \phi'$
3. for necessitation,  $\{\phi\} \vdash_{\text{LiP}} M :_a^C \phi$
4. for epistemic antitonicity,  $\{a \mathbf{k} M \rightarrow a \mathbf{k} M'\} \vdash_{\text{LiP}} (M' :_a^C \phi) \rightarrow M :_a^C \phi$ .

(In the space-saving, horizontal Hilbert-notation “ $\Phi \vdash_{\text{LiP}} \phi$ ”,  $\Phi$  is not a set of hypotheses but a set of premises, see for example modus ponens, necessitation, and epistemic antitonicity.<sup>10</sup>)

<sup>10</sup>So for example *modus ponens* can be presented on one line and even in-line as  $\{\phi, \phi \rightarrow \phi'\} \vdash_{\text{LiP}} \phi'$  rather than on two display lines as

$$\frac{\phi \quad \phi \rightarrow \phi'}{\phi'}.$$

Then  $\vdash_{\text{LiP}}$  can be viewed as being defined by a Cl-induced Hilbert-style proof system. In fact  $\text{Cl} : 2^{\mathcal{L}} \rightarrow 2^{\mathcal{L}}$  is a standard consequence operator, i.e., a substitution-invariant compact closure operator:

1.  $\Gamma \subseteq \text{Cl}(\Gamma)$  (extensivity)
2. if  $\Gamma \subseteq \Gamma'$  then  $\text{Cl}(\Gamma) \subseteq \text{Cl}(\Gamma')$  (monotonicity)
3.  $\text{Cl}(\text{Cl}(\Gamma)) \subseteq \text{Cl}(\Gamma)$  (idempotency)
4.  $\text{Cl}(\Gamma) = \bigcup_{\Gamma' \in 2^{\Gamma}_{\text{finite}}} \text{Cl}(\Gamma')$  (compactness)
5.  $\sigma[\text{Cl}(\Gamma)] \subseteq \text{Cl}(\sigma[\Gamma])$  (substitution invariance),

where  $\sigma$  designates an arbitrary propositional  $\mathcal{L}$ -substitution.

*Proof.* That a Hilbert-style proof system can be viewed as induced by a compact closure operator is well-known (e.g., see [Gab95]); that Cl is indeed such an operator can be verified by inspection of the inductive definition of Cl; and substitution invariance follows from our definitional use of axiom *schemas*.<sup>11</sup>  $\square$

We are going to present some useful (further-used), deducible *structural* laws of LiP, including the deducible non-structural rule of epistemic bitonicity, used in the deduction of some of them. Here, “structural” means “deduced exclusively from term axioms.” The laws are enumerated in a (total) order that respects (but cannot reflect) their respective proof prerequisites.

**Theorem 1** (Some useful deducible structural laws).

1.  $\vdash_{\text{LiP}} a \mathbf{k} (M, M') \rightarrow a \mathbf{k} M$  (left projection, 1-way K-combinator property)
2.  $\vdash_{\text{LiP}} a \mathbf{k} (M, M') \rightarrow a \mathbf{k} M'$  (right projection)
3.  $\vdash_{\text{LiP}} a \mathbf{k} (M, M) \leftrightarrow a \mathbf{k} M$  (pairing idempotency)
4.  $\vdash_{\text{LiP}} a \mathbf{k} (M, M') \leftrightarrow a \mathbf{k} (M', M)$  (pairing commutativity)
5.  $\vdash_{\text{LiP}} (a \mathbf{k} M \rightarrow a \mathbf{k} M') \leftrightarrow (a \mathbf{k} (M, M') \leftrightarrow a \mathbf{k} M)$  (neutral pair elements)
6.  $\vdash_{\text{LiP}} a \mathbf{k} (M, a) \leftrightarrow a \mathbf{k} M$  (self-neutral pair element)
7.  $\vdash_{\text{LiP}} a \mathbf{k} (M, (M', M'')) \leftrightarrow a \mathbf{k} ((M, M'), M'')$  (pairing associativity)
8.  $\{a \mathbf{k} M \leftrightarrow a \mathbf{k} M'\} \vdash_{\text{LiP}} (M :_a^{\mathcal{C}} \phi) \leftrightarrow M' :_a^{\mathcal{C}} \phi$  (epistemic bitonicity)
9.  $\vdash_{\text{LiP}} (M :_a^{\mathcal{C}} \phi) \rightarrow (M', M) :_a^{\mathcal{C}} \phi$  (proof extension, left)
10.  $\vdash_{\text{LiP}} (M :_a^{\mathcal{C}} \phi) \rightarrow (M, M') :_a^{\mathcal{C}} \phi$  (proof extension, right)
11.  $\vdash_{\text{LiP}} ((M :_a^{\mathcal{C}} \phi) \vee M' :_a^{\mathcal{C}} \phi) \rightarrow (M, M') :_a^{\mathcal{C}} \phi$  (proof extension)

<sup>11</sup>Alternatively to axiom schemas, we could have used axioms together with an additional substitution-rule set  $\{ \sigma[\phi] \mid \phi \in \text{Cl}^n(\Gamma) \}$  in the definiens of  $\text{Cl}^{n+1}(\Gamma)$ .

12.  $\vdash_{\text{LiP}} ((M, M) :_a^C \phi) \leftrightarrow M :_a^C \phi$  (*proof idempotency*)
13.  $\vdash_{\text{LiP}} ((M, M') :_a^C \phi) \leftrightarrow (M', M) :_a^C \phi$  (*proof commutativity*)
14.  $\{a \mathbf{k} M \rightarrow a \mathbf{k} M'\} \vdash_{\text{LiP}} ((M, M') :_a^C \phi) \leftrightarrow M :_a^C \phi$  (*neutral proof elements*)
15.  $\vdash_{\text{LiP}} ((M, a) :_a^C \phi) \leftrightarrow M :_a^C \phi$  (*self-neutral proof element*)
16.  $\vdash_{\text{LiP}} ((M, (M', M'')) :_a^C \phi) \leftrightarrow ((M, M'), M'') :_a^C \phi$  (*proof associativity*)
17.  $\vdash_{\text{LiP}} (\llbracket M \rrbracket_a :_a^C \phi) \rightarrow M :_a^C \phi$  (*self-signing elimination*)
18.  $\vdash_{\text{LiP}} ((M :_a^C \phi) \vee b :_a^C \phi) \rightarrow \llbracket M \rrbracket_b :_a^C \phi$  (*signing introduction*)
19.  $\vdash_{\text{LiP}} (\llbracket M \rrbracket_a :_a^C \phi) \leftrightarrow M :_a^C \phi$  (*self-signing idempotency*)
20. When  $\mathcal{A} = \{a\}$  (*singleton society*):
  - (a)  $\vdash_{\text{LiP}} a \mathbf{k} M$  (*total knowledge*)
  - (b)  $\vdash_{\text{LiP}} a \mathbf{k} M \leftrightarrow a \mathbf{k} M'$  (*epistemic indifference*)
  - (c)  $\vdash_{\text{LiP}} (M :_a^C \phi) \leftrightarrow M' :_a^C \phi$  (*proof indifference*).

*Proof.* See Appendix B.1. □

Proof extension and idempotency jointly define *proof redundancy*. Then, for the cases where  $\mathcal{A} = \{a\}$ , bear in mind that  $\mathcal{M}$  is a function of  $\mathcal{A}$ , and that LiP has been designed for truly interactive cases, i.e., cases where  $|\mathcal{A}| > 1$  and *not* for non-interactive or degenerately interactive cases, i.e., cases where  $|\mathcal{A}| = 1$ . So **when**  $\mathcal{A} = \{a\}$ ,  $\mathcal{M}$  **is actually strictly smaller than when**  $\mathcal{A} \supsetneq \{a\}$ ! In particular when  $|\mathcal{A}| > 1$ , obviously neither total knowledge nor epistemic indifference holds, nor does proof indifference hold. For the fortunate failure of proof indifference when  $|\mathcal{A}| > 1$ , consider the following doubly minimal counter-example. Without loss of generality, let  $\mathcal{A} := \{a, b\}$  such that  $a \neq b$ . Then  $\vdash_{\text{LiP}} b :_a^0 a \mathbf{k} b$  (instance of self-knowledge, cf. Theorem 2), but  $\not\vdash_{\text{LiP}} a :_a^0 a \mathbf{k} b$  intuitively, and also formally. Just imagine a state in which  $a$  does not know  $b$ 's name string, cf. Definition 3 and 4; then giving her her own name string, which she already knows anyway, will not make her know  $b$ 's; and then apply the contraposition of axiomatic soundness, cf. Theorem 5. The counter-example is doubly minimal in the sense that both the involved proof terms ( $a$  and  $b$  are atomic terms) as well as the involved proof goal ( $a \mathbf{k} b$  is an atomic proposition about atomic terms) are minimal. Note that we could of course conceive LiP without the  $a \mathbf{k} a$ -axiom for some or even all  $a \in \mathcal{A}$  and arbitrary  $\mathcal{A}$ . In particular when  $\mathcal{A} = \{a\}$ , excluding  $a \mathbf{k} a$  from  $\Gamma_1$  definitely makes sense, since agent names really make sense only for non-empty non-singleton societies. In such a system, say  $\text{LiP}^-$ , obviously none of the singleton-society laws of LiP would hold for  $a$ , and thus also non-interactive, singleton-society examples (e.g., Kripke's Red Barn Example in [Art08a]) could be faithfully formalised. The price to pay for  $\text{LiP}^-$  would be, first, the failure of the following laws: self-neutral pair element, self-neutral proof element, and, cf. Theorem 2, self-truthfulness, the left implication

of self-truthfulness *bis*, own name strings cannot prove falsehood, and own name strings are consistent proofs; and thus, second, the impoverishment of the proof-term structure from an idempotent commutative monoid (cf. Corollary 4) to an idempotent commutative semigroup (loss of the neutral element). (The failure of these laws does *not* imply that their negation succeeds, because LiP-like theories are negation-*incomplete*, cf. Section 2.3.) However, how this price is appreciated eventually depends on the considered application. For example, the failure of self-truthfulness could even be considered desirable: if we were to exclude  $a \mathbf{k} a$  from  $\Gamma_1$ , we would actually exclude  $(M :_a^C \phi) \rightarrow \phi$  from being a theorem in the resulting logical system  $\text{LiP}^-$  for *all*  $M \in \mathcal{M}$ , like in the Gödel-Löb Logic of (non-interactive) Provability GL [JdJ98, AB05]. Next, the 1-way K-combinator property and the following simple corollary of Theorem 1 jointly establish the fact that our agents can be viewed as combinators in the sense of combinatory logic (CL) viewed as a (non-equational) theory of term reduction [HS08]. (The converse of the K-combinator property does not hold.)

**Corollary 1** (S-combinator property).

1.  $\vdash_{\text{LiP}} a \mathbf{k} ((M, M'), M'') \leftrightarrow a \mathbf{k} (M, (M'', (M', M'')))$
2.  $\vdash_{\text{LiP}} (((M, M'), M'') :_a^C \phi) \leftrightarrow (M, (M'', (M', M''))) :_a^C \phi$

*Proof.* 1 follows jointly from idempotency (copy  $M'''$ ), commutativity, and associativity of pairing; and 2 follows jointly from 1 and epistemic bitonicity.  $\square$

Note that thanks to the modular set-up of LiP, epistemic antitonicity would equally easily yield the application-specific modal laws for:

- hashing:  $([M] :_a^C \phi) \rightarrow M :_a^C \phi$
- encryption:  $([M]_{M'} :_a^C \phi) \rightarrow (M, M') :_a^C \phi$
- decryption:  $(M :_a^C \phi) \rightarrow ([M]_{M'}, M') :_a^C \phi$
- plugging:  $(M[M'/x] :_a^C \phi) \rightarrow (M(x), M') :_a^C \phi$
- unplugging:  $(M(x) :_a^C \phi) \rightarrow M[M'/x] :_a^C \phi$ .

We are going to present also some useful (further-used), deducible *logical* laws of LiP. Here, “logical” means “not structural” in the previously defined sense. Also these laws are enumerated in an order that respects their respective proof prerequisites.

**Theorem 2** (Some useful deducible logical laws).

1.  $\vdash_{\text{LiP}} (M :_a^C (\phi \rightarrow \phi')) \rightarrow ((M :_a^C \phi) \rightarrow M :_a^C \phi') \quad (\text{Kripke's law, } K)$
2.  $\{\phi \rightarrow \phi'\} \vdash_{\text{LiP}} (M :_a^C \phi) \rightarrow M :_a^C \phi' \quad (\text{regularity})$
3.  $\{\phi \leftrightarrow \phi'\} \vdash_{\text{LiP}} (M :_a^C \phi) \leftrightarrow M :_a^C \phi' \quad (\text{regularity bis})$



4.  $\{a \mathbf{k} M \rightarrow a \mathbf{k} M', \phi \rightarrow \phi'\} \vdash_{\text{LiP}} (M' :_a^{\mathcal{C}} \phi) \rightarrow M :_a^{\mathcal{C}} \phi' \quad (\text{epistemic regularity})$
5.  $\{a \mathbf{k} M \leftrightarrow a \mathbf{k} M', \phi \leftrightarrow \phi'\} \vdash_{\text{LiP}} (M' :_a^{\mathcal{C}} \phi) \leftrightarrow M :_a^{\mathcal{C}} \phi' \quad (\text{epistemic regularity bis})$
6.  $\vdash_{\text{LiP}} ((M :_a^{\mathcal{C}} \phi) \wedge M' :_a^{\mathcal{C}} \phi') \rightarrow (M, M') :_a^{\mathcal{C}} (\phi \wedge \phi') \quad (\text{proof conjunctions})$
7.  $\vdash_{\text{LiP}} ((M :_a^{\mathcal{C}} \phi) \wedge M :_a^{\mathcal{C}} \phi') \leftrightarrow M :_a^{\mathcal{C}} (\phi \wedge \phi') \quad (\text{proof conjunctions bis})$
8.  $\vdash_{\text{LiP}} ((M :_a^{\mathcal{C}} \phi) \vee M' :_a^{\mathcal{C}} \phi') \rightarrow (M, M') :_a^{\mathcal{C}} (\phi \vee \phi') \quad (\text{proof disjunctions})$
9.  $\vdash_{\text{LiP}} ((M :_a^{\mathcal{C}} \phi) \vee M :_a^{\mathcal{C}} \phi') \rightarrow M :_a^{\mathcal{C}} (\phi \vee \phi') \quad (\text{proof disjunctions bis})$
10.  $\vdash_{\text{LiP}} M :_a^{\mathcal{C}} \top \quad (\text{anything can prove tautological truth})$
11.  $\vdash_{\text{LiP}} (a :_a^{\mathcal{C}} \phi) \rightarrow \phi \quad (\text{self-truthfulness})$
12.  $\phi \dashv\vdash_{\text{LiP}} a :_a^{\mathcal{C}} \phi \quad (\text{self-truthfulness bis})$
13.  $\vdash_{\text{LiP}} a \mathbf{k} M \rightarrow \neg(M :_a^{\mathcal{C}} \perp) \quad (\text{nothing known can prove falsehood})$
14.  $\vdash_{\text{LiP}} \neg(a :_a^{\mathcal{C}} \perp) \quad (\text{own name strings cannot prove falsehood})$
15.  $\vdash_{\text{LiP}} a \mathbf{k} M \rightarrow ((M :_a^{\mathcal{C}} \phi) \rightarrow M \diamond_a^{\mathcal{C}} \phi) \quad (\text{epistemic proof consistency})$
16.  $\vdash_{\text{LiP}} (a :_a^{\mathcal{C}} \phi) \rightarrow a \diamond_a^{\mathcal{C}} \phi \quad (\text{own name strings are consistent proofs})$
17.  $\vdash_{\text{LiP}} \llbracket M \rrbracket_b :_a^{\mathcal{C} \cup \{b\}} b \mathbf{k} M \quad (\text{authentic knowledge})$
18.  $\vdash_{\text{LiP}} M :_a^{\emptyset} a \mathbf{k} M \quad (\text{self-knowledge})$
19.  $\vdash_{\text{LiP}} (M :_a^{\mathcal{C}} \phi) \rightarrow \bigwedge_{b \in \mathcal{C} \cup \{a\}} (\llbracket M \rrbracket_a :_b^{\mathcal{C} \cup \{a\}} \phi) \quad (\text{simple peer review})$
20.  $\vdash_{\text{LiP}} (M :_a^{\mathcal{C} \cup \mathcal{C}'} \phi) \rightarrow ((M :_a^{\mathcal{C}} \phi) \wedge M :_a^{\mathcal{C}'} \phi) \quad (\text{group decomposition bis})$
21.  $\vdash_{\text{LiP}} (M :_a^{\mathcal{C} \cup \{a\}} \phi) \leftrightarrow (M :_a^{\mathcal{C}} \phi) \quad (\text{self-neutral group element})$
22.  $\vdash_{\text{LiP}} M :_a^{\mathcal{C}} ((M :_a^{\mathcal{C}} \phi) \rightarrow \phi) \quad (\text{self-proof of truthfulness})$
23.  $\vdash_{\text{LiP}} M :_a^{\mathcal{C}} (\neg(M :_a^{\mathcal{C}} \perp)) \quad (\text{self-proof of proof consistency})$
24.  $\vdash_{\text{LiP}} (M :_a^{\mathcal{C}} \phi) \rightarrow M :_a^{\mathcal{C}} (\bigwedge_{b \in \mathcal{C} \cup \{a\}} (\llbracket M \rrbracket_a :_b^{\mathcal{C} \cup \{a\}} \phi)) \quad (\text{simple peer review bis})$
25.  $\vdash_{\text{LiP}} (M :_a^{\mathcal{C}} (M :_a^{\mathcal{C}} \phi)) \leftrightarrow M :_a^{\mathcal{C}} \phi \quad (\text{modal idempotency})$
26. When  $\mathcal{A} = \{a\}$  (singleton society):
  - (a)  $\vdash_{\text{LiP}} \neg(M :_a^{\mathcal{C}} \perp) \quad (\text{nothing can prove falsehood})$
  - (b)  $\vdash_{\text{LiP}} (M :_a^{\mathcal{C}} \phi) \rightarrow \phi \quad (\text{truthfulness})$
  - (c)  $\vdash_{\text{LiP}} (M :_a^{\mathcal{C}} \phi) \rightarrow M \diamond_a^{\mathcal{C}} \phi \quad (\text{proof consistency}).$

*Proof.* See Appendix B.2.  $\square$

Kripke's law and the law of modal idempotency are discussed in Section 3.2.2. The key to their validity is that LiP-agents are resource-unbounded (though are unable to guess) and act themselves as proof checkers (no need for LP's '!'). Notice that regularity and epistemic antitonicity resemble each other in that both laws relate an implicational premise with an implicational conclusion about proof modalities, but while regularity relates the modality operands monotonically, epistemic antitonicity relates the proof parameters antitonically. Both laws are combined in the law of epistemic regularity. The law that nothing known can prove falsehood and the law of epistemic proof consistency, which both result through proof from epistemic truthfulness, might raise doubt about the consistency of LiP. If so, Corollary 3 will dispel this doubt.

**Corollary 2** (Normality). *LiP is a normal modal logic.*

*Proof.* Jointly by Kripke's law (cf. Theorem 2), *modus ponens* (by definition), necessitation (by definition), and substitution invariance (cf. Proposition 1).  $\square$

In contrast, LP is, technically speaking, not a normal modal logic [Art07, Section 5].

**Definition 3** (Semantic ingredients). For the constructive model-theoretic study of LiP, let

- $\mathcal{S}$  designate the *state space*—a set of *system states*  $s$
- $\text{msgs}_a : \mathcal{S} \rightarrow 2^{\mathcal{M}}$  designate a *raw-data extractor* that extracts (without analysing) the (finite) set of messages from a system state  $s$  that  $a$  has either generated (assuming that only  $a$  can generate  $a$ 's signature) or else received *as such* (not only as a strict subterm of another message); that is,  $\text{msgs}_a(s)$  is  $a$ 's *data base* in  $s$
- $\text{cl}_a^s : 2^{\mathcal{M}} \rightarrow 2^{\mathcal{M}}$  designate a *data-mining operator* such that  $\text{cl}_a^s(\mathcal{D}) := \text{cl}_a(\text{msgs}_a(s) \cup \mathcal{D}) := \bigcup_{n \in \mathbb{N}} \text{cl}_a^n(\text{msgs}_a(s) \cup \mathcal{D})$ , where for all  $\mathcal{D} \subseteq \mathcal{M}$ :

$$\begin{aligned} \text{cl}_a^0(\mathcal{D}) &:= \{a\} \cup \mathcal{D} \\ \text{cl}_a^{n+1}(\mathcal{D}) &:= \text{cl}_a^n(\mathcal{D}) \cup \\ &\quad \{ (M, M') \mid \{M, M'\} \subseteq \text{cl}_a^n(\mathcal{D}) \} \cup \quad (\text{pairing}) \\ &\quad \{ M, M' \mid (M, M') \in \text{cl}_a^n(\mathcal{D}) \} \cup \quad (\text{unpairing}) \\ &\quad \{ \llbracket M \rrbracket_a \mid M \in \text{cl}_a^n(\mathcal{D}) \} \cup \quad (\text{personal signature synthesis}) \\ &\quad \{ M \mid \llbracket M \rrbracket_b \in \text{cl}_a^n(\mathcal{D}) \} \quad (\text{universal signature analysis}) \end{aligned}$$

( $\text{cl}_a^s(\emptyset)$  can be viewed as  $a$ 's *individual-knowledge base* in  $s$ . For application-specific terms such as encryption, we would have to add here the closure conditions corresponding to their characteristic term axioms.)

- $\leq_a \subseteq \mathcal{S} \times \mathcal{S}$  designate a *data preorder* on states such that for all  $s, s' \in \mathcal{S}$ ,  $s \leq_a s' : \text{iff } \text{cl}_a^s(\emptyset) \subseteq \text{cl}_a^{s'}(\emptyset)$

(The reader is invited to consider the effects of encryption on closure here.)

- $\leq_c := (\bigcup_{a \in \mathcal{C}} \leq_a)^*$ , where ‘ $*$ ’ designates the Kleene (i.e., the reflexive transitive) closure operation on binary relations
- $\equiv_a := \leq_a \cap (\leq_a)^{-1}$  designate an equivalence relation of *state indistinguishability*, where ‘ $^{-1}$ ’ designates the converse operation on binary relations
- ${}_M R_a^C \subseteq \mathcal{S} \times \mathcal{S}$  designate our *concretely constructed accessibility relation*—short, *concrete accessibility*—for the proof modality so that for all  $s, s' \in \mathcal{S}$ ,

$$s {}_M R_a^C s' \text{ :iff } \begin{array}{l} s' \in \bigcup [\check{s}]_{\equiv_a} \\ s \leq_{\mathcal{C} \cup \{a\}} \check{s} \text{ and} \\ M \in \text{cl}_a^{\check{s}}(\emptyset) \end{array} \quad (1)$$

(iff there is  $\check{s} \in \mathcal{S}$  s.t.  $s \leq_{\mathcal{C} \cup \{a\}} \check{s}$  and  $M \in \text{cl}_a^{\check{s}}(\emptyset)$  and  $\check{s} \equiv_a s'$ )

(See Section 2.1 for an extensive explication of this elementary construction.)

**Proposition 2** (Data closure).  $\text{cl}_a : 2^{\mathcal{M}} \rightarrow 2^{\mathcal{M}}$  is a compact closure operator:

1.  $\mathcal{D} \subseteq \text{cl}_a(\mathcal{D})$  (*extensivity*)
2. if  $\mathcal{D} \subseteq \mathcal{D}'$  then  $\text{cl}_a(\mathcal{D}) \subseteq \text{cl}_a(\mathcal{D}')$  (*monotonicity*)
3.  $\text{cl}_a(\text{cl}_a(\mathcal{D})) \subseteq \text{cl}_a(\mathcal{D})$  (*idempotency*)
4.  $\text{cl}_a(\mathcal{D}) = \bigcup_{\mathcal{D}' \in 2_{\text{finite}}^{\mathcal{P}}} \text{cl}_a(\mathcal{D}')$  (*compactness*),

*Proof.* By inspection of the inductive definition of  $\text{cl}_a$ . □

The operator  $\text{cl}_a$  induces a relation  $\vdash_a \subseteq 2^{\mathcal{M}} \times \mathcal{M}$  of *data derivation* such that

$$\mathcal{D} \vdash_a M \text{ :iff } M \in \text{cl}_a(\mathcal{D}).$$

Hence, an agent  $a$  can be viewed as a *data miner* who mines the data  $\mathcal{D}$  by means of the [SD08, *association*] *rules* for pairing and signing (and possibly other, application-specific constructors) that define the closure operator  $\text{cl}_a$ .

**Proposition 3** (Data derivation).

**Cut** If  $\mathcal{D} \vdash_a M$  and  $\{M\} \vdash_a M'$  then  $\mathcal{D} \vdash_a M'$ .

**Compactness** If  $\mathcal{D} \vdash_a M$  then there is a finite  $\mathcal{D}' \subseteq \mathcal{D}$  such that  $\mathcal{D}' \vdash_a M$ .

**Complexity** For all finite  $\mathcal{D} \subseteq \mathcal{M}$ , “ $\mathcal{D} \vdash_a M$ ” is decidable in deterministic polynomial time in the size of  $\mathcal{D}$  and  $M$ .

**Connection to Scott information systems** Let for all  $a \in \mathcal{A}$ ,  $s \in \mathcal{S}$ , and  $\mathcal{D} \subseteq \mathcal{M}$ ,

$$\mathcal{C}_a^s(\mathcal{D}) := \{ \mathcal{D}' \subseteq \mathcal{D} \mid \text{cl}_a^s(\mathcal{D}') = \mathcal{D}' \}.$$

Further, let

$$\text{Con}_a^s := \bigcup_{\mathcal{D} \in \mathcal{C}_a^s(\mathcal{M})} 2_{\text{finite}}^{\mathcal{D}}.$$

Then,

$$\langle \mathcal{M}, \text{Con}_a^s, \vdash_a \rangle$$

is a Scott information system, i.e., for all  $M \in \mathcal{M}$ ,  $\mathcal{D} \in \text{Con}_a^s$ , and  $\mathcal{D}' \subseteq \mathcal{M}$ :

1.  $\{M\} \in \text{Con}_a^s$
2. if  $M \in \mathcal{D}$  then  $\mathcal{D} \vdash_a M$
3. if  $\mathcal{D}' \subseteq \mathcal{D}$  then  $\mathcal{D}' \in \text{Con}_a^s$
4. if  $\mathcal{D} \vdash_a M$  then  $\mathcal{D} \cup \{M\} \in \text{Con}_a^s$
5. if  $\mathcal{D}' \in \text{Con}_a^s$  and  $\mathcal{D} \vdash_a \mathcal{D}'$  and  $\mathcal{D}' \vdash_a M$  then  $\mathcal{D} \vdash_a M$ , where  $\mathcal{D} \vdash_a \mathcal{D}'$  :iff for all  $M' \in \mathcal{D}'$ ,  $\mathcal{D} \vdash_a M'$ .

(Message terms are information tokens in the sense of Dana Scott [DP02, Chapter 9].)

*Proof.* The cut and the compactness property follow by inspection of the defining cases of  $\text{cl}_a^s$ . The complexity follows from the complexity of message derivation for even more complex message languages (e.g., including encryption and other constructors [TGD10]). Regarding the connection to Scott information systems: Property 1 follows from the fact that  $\{M\} \in 2_{\text{finite}}^{\mathcal{M}}$  and  $\mathcal{M} \in \mathcal{C}_a^s(\mathcal{M})$ , Property 2 from the definition of  $\vdash_a$ , Property 3 from the powerset construction, Property 4 from the definition of  $\vdash_a$ , and Property 5 jointly from the finiteness of  $\mathcal{D}'$  (which can be transformed into a message pair [of pairs]) and the cut property of  $\vdash_a$ .  $\square$

**Proposition 4** (Concrete accessibility).

1. If  $M \in \text{cl}_a^s(\emptyset)$  then  $s \text{MR}_a^C s$  (conditional reflexivity).
2. For all  $\mathcal{C}' \subseteq \mathcal{A}$ , if  $\mathcal{C} \subseteq \mathcal{C}'$  then  $\text{MR}_a^C \subseteq \text{MR}_a^{\mathcal{C}'}$  (communal monotonicity).
3. If  $s \text{MR}_a^C s'$  and  $s' \text{MR}_a^C s''$  then  $s \text{MR}_a^C s''$  (transitivity).

*Proof.* For 1, let  $s \in \mathcal{S}$  and suppose that  $M \in \text{cl}_a^s(\emptyset)$ . Further,  $s \leq_{\mathcal{C} \cup \{a\}} s$  and  $s \equiv_a s$ , by reflexivity. Hence  $s \text{MR}_a^C s$ . For 2, let  $\mathcal{C}' \subseteq \mathcal{A}$  and suppose that  $\mathcal{C} \subseteq \mathcal{C}'$ . Further, let  $s, s' \in \mathcal{S}$  and suppose that  $s \text{MR}_a^C s'$ . That is, there is  $\check{s} \in \mathcal{S}$  such that  $s \leq_{\mathcal{C} \cup \{a\}} \check{s}$  and  $M \in \text{cl}_a^{\check{s}}(\emptyset)$  and  $\check{s} \equiv_a s'$ . Hence  $s \leq_{\mathcal{C}' \cup \{a\}} \check{s}$ , and thus  $s \text{MR}_a^{\mathcal{C}'} s'$ . For 3, let  $s, s', s'' \in \mathcal{S}$  and suppose that  $s \text{MR}_a^C s'$  and  $s' \text{MR}_a^C s''$ . That is, there is  $\check{s} \in \mathcal{S}$  such that  $s \leq_{\mathcal{C} \cup \{a\}} \check{s}$  and  $M \in \text{cl}_a^{\check{s}}(\emptyset)$  and  $\check{s} \equiv_a s'$  (thus

$\check{s} \leq_{\mathcal{C} \cup \{a\}} s')$ , and there is  $\check{s}' \in \mathcal{S}$  such that  $s' \leq_{\mathcal{C} \cup \{a\}} \check{s}'$  and  $M \in \text{cl}_a^{\check{s}'}(\emptyset)$  and  $\check{s}' \equiv_a s''$ . Hence  $s \leq_{\mathcal{C} \cup \{a\}} s'$  and then  $s \leq_{\mathcal{C} \cup \{a\}} \check{s}'$ , both by transitivity, and thus  $s \mathrel{M\mathcal{R}_a^{\mathcal{C}}} s''$ .  $\square$

**Definition 4** (Kripke-model). We define the *satisfaction relation*  $\models$  for LiP such that:

$$\begin{aligned} (\mathfrak{S}, \mathcal{V}), s &\models P && \text{iff } s \in \mathcal{V}(P) \\ (\mathfrak{S}, \mathcal{V}), s &\models \neg \phi && \text{iff not } (\mathfrak{S}, \mathcal{V}), s \models \phi \\ (\mathfrak{S}, \mathcal{V}), s &\models \phi \wedge \phi' && \text{iff } (\mathfrak{S}, \mathcal{V}), s \models \phi \text{ and } (\mathfrak{S}, \mathcal{V}), s \models \phi' \\ (\mathfrak{S}, \mathcal{V}), s &\models M :_a^{\mathcal{C}} \phi && \text{iff for all } s' \in \mathcal{S}, \text{ if } s \mathrel{M\mathcal{R}_a^{\mathcal{C}}} s' \text{ then } (\mathfrak{S}, \mathcal{V}), s' \models \phi, \end{aligned}$$

where

- $\mathcal{V} : \mathcal{P} \rightarrow 2^{\mathcal{S}}$  designates a usual *valuation function*, yet partially predefined such that for all  $a \in \mathcal{A}$  and  $M \in \mathcal{M}$ ,

$$\mathcal{V}(a \mathbf{k} M) := \{ s \in \mathcal{S} \mid M \in \text{cl}_a^s(\emptyset) \}$$

(If agents are Turing-machines then  $a$  knowing  $M$  can be understood as  $a$  being able to parse  $M$  on its tape.)

- $\mathfrak{S} := (\mathcal{S}, \{M\mathcal{R}_a^{\mathcal{C}}\}_{M \in \mathcal{M}, a \in \mathcal{A}, \mathcal{C} \subseteq \mathcal{A}})$  designates a (modal) *frame* for LiP with (in analogy to LP) an *abstractly constrained accessibility relation*— short, *abstract accessibility*— $M\mathcal{R}_a^{\mathcal{C}} \subseteq \mathcal{S} \times \mathcal{S}$  for the proof modality such that

– (a *priori* constraints):

- \* if  $M \in \text{cl}_a^s(\emptyset)$  then  $s \mathrel{M\mathcal{R}_a^{\mathcal{C}}} s$
- \* for all  $\mathcal{C}' \subseteq \mathcal{A}$ , if  $\mathcal{C} \subseteq \mathcal{C}'$  then  $M\mathcal{R}_a^{\mathcal{C}} \subseteq M\mathcal{R}_a^{\mathcal{C}'}$
- \* if  $s \mathrel{M\mathcal{R}_a^{\mathcal{C}}} s'$  and  $s' \mathrel{M\mathcal{R}_a^{\mathcal{C}}} s''$  then  $s \mathrel{M\mathcal{R}_a^{\mathcal{C}}} s''$

– (a *posteriori* constraints):

- \* if (for all  $s' \in \mathcal{S}$ ,  $M \in \text{cl}_a^{s'}(\emptyset)$  implies  $M' \in \text{cl}_a^{s'}(\emptyset)$ ) then  ${}_s\llbracket M' \rrbracket_a^{\mathcal{C}} \subseteq {}_s\llbracket M \rrbracket_a^{\mathcal{C}}$
- \* if  $(\phi \rightarrow \phi') \in {}_s\llbracket M \rrbracket_a^{\mathcal{C}}$  and  $\phi \in {}_s\llbracket M' \rrbracket_a^{\mathcal{C}}$  then  $\phi' \in {}_s\llbracket (M, M') \rrbracket_a^{\mathcal{C}}$
- \* if  $\phi \in {}_s\llbracket M \rrbracket_a^{\mathcal{C}}$  then for all  $b \in \mathcal{C} \cup \{a\}$ ,  $(a \mathbf{k} M \wedge M :_a^{\mathcal{C}} \phi) \in {}_s\llbracket \{M\} \rrbracket_{ab}^{\mathcal{C} \cup \{a\}}$

where *message meaning*  ${}_s\llbracket \cdot \rrbracket_a^{\mathcal{C}} : \mathcal{M} \rightarrow 2^{\mathcal{L}}$  is defined as

$${}_s\llbracket M \rrbracket_a^{\mathcal{C}} := \{ \phi \in \mathcal{L} \mid (\mathfrak{S}, \mathcal{V}), s \models M :_a^{\mathcal{C}} \phi \}$$

- $(\mathfrak{S}, \mathcal{V})$  designates a (modal) *model* for LiP.

Notice that message meaning contains *agent meaning*  ${}_s\llbracket b \rrbracket_a^C$  (agent names are particular messages) in the sense that the meaning of the agent  $b$  to the community  $\mathcal{C} \cup \{a\}$  in the state  $s$  is what  $b$ 's name (i.e.,  $b$ ) can prove to  $\mathcal{C} \cup \{a\}$ .

Now, Proposition 4 and 5 jointly establish the important fact that our concrete accessibility in Definition 3 realises all the required properties of our abstract accessibility in Definition 4.

**Proposition 5** (Semantic interface).

1. For all  $s, s' \in \mathcal{S}$ , if  $s \mathrel{MR}_a^C s'$  then  ${}_s\llbracket M \rrbracket_a^C \subseteq {}_{s'}\llbracket M \rrbracket_a^C$ .
2. Let  ${}_M\mathcal{R}_a^C := {}_M\mathcal{R}_a^C$ . Then for all  $s \in \mathcal{S}$ :
  - (a) if (for all  $s' \in \mathcal{S}$ ,  $M \in \text{cl}_a^{s'}(\emptyset)$  implies  $M' \in \text{cl}_a^{s'}(\emptyset)$ ) then  ${}_s\llbracket M' \rrbracket_a^C \subseteq {}_s\llbracket M \rrbracket_a^C$
  - (b) if  $(\phi \rightarrow \phi') \in {}_s\llbracket M \rrbracket_a^C$  and  $\phi \in {}_s\llbracket M' \rrbracket_a^C$  then  $\phi' \in {}_s\llbracket (M, M') \rrbracket_a^C$
  - (c) if  $\phi \in {}_s\llbracket M \rrbracket_a^C$  then for all  $b \in \mathcal{C} \cup \{a\}$ ,  $(a \mathrel{k} M \wedge M :_a^C \phi) \in {}_s\llbracket \{M\} \rrbracket_b^{C \cup \{a\}}$ .

*Proof.* For 1, let  $s, s' \in \mathcal{S}$  and suppose that  $s \mathrel{MR}_a^C s'$ . Further, let  $\phi \in \mathcal{L}$  and suppose that  $\phi \in {}_s\llbracket M \rrbracket_a^C$ , i.e., for all  $s' \in \mathcal{S}$ , if  $s \mathrel{MR}_a^C s'$  then  $(\mathfrak{S}, \mathcal{V}), s' \models \phi$ . Furthermore, let  $s'' \in \mathcal{S}$  and suppose that  $s' \mathrel{MR}_a^C s''$ . Hence  $s \mathrel{MR}_a^C s''$  by transitivity, and thus  $(\mathfrak{S}, \mathcal{V}), s'' \models \phi$ .

For the rest, let  ${}_M\mathcal{R}_a^C := {}_M\mathcal{R}_a^C$  and let  $s \in \mathcal{S}$ .

For 2.a, suppose that for all  $s' \in \mathcal{S}$ ,  $M \in \text{cl}_a^{s'}(\emptyset)$  implies  $M' \in \text{cl}_a^{s'}(\emptyset)$ . Further, let  $\phi \in \mathcal{L}$  and suppose that  $\phi \in {}_s\llbracket M' \rrbracket_a^C$ . That is, for all  $s' \in \mathcal{S}$ , if (there is  $\check{s} \in \mathcal{S}$  such that  $s \leq_{\mathcal{C} \cup \{a\}} \check{s}$  and  $M' \in \text{cl}_a^{\check{s}}(\emptyset)$  and  $\check{s} \equiv_a s'$ ) then  $(\mathfrak{S}, \mathcal{V}), s' \models \phi$ . Furthermore, suppose that (there is  $\check{s} \in \mathcal{S}$  such that  $s \leq_{\mathcal{C} \cup \{a\}} \check{s}$  and  $M \in \text{cl}_a^{\check{s}}(\emptyset)$  and  $\check{s} \equiv_a s'$ ). Hence  $M' \in \text{cl}_a^{\check{s}}(\emptyset)$  by the first hypothesis, and thus  $(\mathfrak{S}, \mathcal{V}), s' \models \phi$ .

For 2.b, suppose that  $(\phi \rightarrow \phi') \in {}_s\llbracket M \rrbracket_a^C$  and  $\phi \in {}_s\llbracket M' \rrbracket_a^C$ . That is: for all  $s' \in \mathcal{S}$ , if (there is  $\check{s} \in \mathcal{S}$  such that  $s \leq_{\mathcal{C} \cup \{a\}} \check{s}$  and  $M \in \text{cl}_a^{\check{s}}(\emptyset)$  and  $\check{s} \equiv_a s'$ ) then  $(\mathfrak{S}, \mathcal{V}), s' \models \phi \rightarrow \phi'$ ; and for all  $s' \in \mathcal{S}$ , if (there is  $\check{s} \in \mathcal{S}$  such that  $s \leq_{\mathcal{C} \cup \{a\}} \check{s}$  and  $M' \in \text{cl}_a^{\check{s}}(\emptyset)$  and  $\check{s} \equiv_a s'$ ) then  $(\mathfrak{S}, \mathcal{V}), s' \models \phi$ . Further, let  $s' \in \mathcal{S}$  and suppose that there is  $\check{s} \in \mathcal{S}$  such that  $s \leq_{\mathcal{C} \cup \{a\}} \check{s}$  and  $(M, M') \in \text{cl}_a^{\check{s}}(\emptyset)$  and  $\check{s} \equiv_a s'$ . Hence  $M \in \text{cl}_a^{\check{s}}(\emptyset)$  and  $M' \in \text{cl}_a^{\check{s}}(\emptyset)$ . Hence  $(\mathfrak{S}, \mathcal{V}), s' \models \phi \rightarrow \phi'$  and  $(\mathfrak{S}, \mathcal{V}), s' \models \phi$ , respectively. Hence  $(\mathfrak{S}, \mathcal{V}), s' \models \phi'$ , and thus  $\phi' \in {}_s\llbracket (M, M') \rrbracket_a^C$ .

For 2.c, see the proof of Theorem 3.  $\square$

Observe that LiP has a Herbrand-style semantics, i.e., logical constants (agent names) and functional symbols (pairing, signing) are self-interpreted rather than interpreted in terms of (other, semantic) constants and functions. This simplifying design choice spares our framework from the additional complexity that would arise from term-variable assignments [BG07], which in turn keeps our models propositionally modal. Our choice is admissible because our individuals (messages) are finite. (Infinitely long “messages” are non-messages; they can

never be completely received, e.g., transmitting irrational numbers as such is impossible.)

**Definition 5** (Truth & Validity).

- The formula  $\phi \in \mathcal{L}$  is *true* (or *satisfied*) in the model  $(\mathfrak{S}, \mathcal{V})$  at the state  $s \in \mathcal{S}$  :iff  $(\mathfrak{S}, \mathcal{V}), s \models \phi$ .
- The formula  $\phi$  is *satisfiable* in the model  $(\mathfrak{S}, \mathcal{V})$  :iff there is  $s \in \mathcal{S}$  such that  $(\mathfrak{S}, \mathcal{V}), s \models \phi$ .
- The formula  $\phi$  is *globally true* (or *globally satisfied*) in the model  $(\mathfrak{S}, \mathcal{V})$ , written  $(\mathfrak{S}, \mathcal{V}) \models \phi$ , :iff for all  $s \in \mathcal{S}$ ,  $(\mathfrak{S}, \mathcal{V}), s \models \phi$ .
- The formula  $\phi$  is *satisfiable* :iff there is a model  $(\mathfrak{S}, \mathcal{V})$  and a state  $s \in \mathcal{S}$  such that  $(\mathfrak{S}, \mathcal{V}), s \models \phi$ .
- The formula  $\phi$  is (*universally true* or) *valid*, written  $\models \phi$ , :iff for all models  $(\mathfrak{S}, \mathcal{V})$ ,  $(\mathfrak{S}, \mathcal{V}) \models \phi$ . (cf. [BvB07])

So we can paraphrase the law of epistemic antitonicity in Definition 2 as: “Whatever a universally poorer message  $M'$  can prove to  $a$ , any universally richer message  $M$  can also prove to  $a$ , and this in all social contexts  $\mathcal{C} \cup \{a\}$ .”

**Proposition 6** (Admissibility of specific axioms and rules).

1.  $\models a \mathbf{k} a$
2.  $\models a \mathbf{k} M \rightarrow a \mathbf{k} \llbracket M \rrbracket_a$
3.  $\models a \mathbf{k} \llbracket M \rrbracket_b \rightarrow a \mathbf{k} (M, b)$
4.  $\models (a \mathbf{k} M \wedge a \mathbf{k} M') \leftrightarrow a \mathbf{k} (M, M')$
5.  $\models (M :_a^{\mathcal{C}} (\phi \rightarrow \phi')) \rightarrow ((M' :_a^{\mathcal{C}} \phi) \rightarrow (M, M') :_a^{\mathcal{C}} \phi')$
6.  $\models (M :_a^{\mathcal{C}} \phi) \rightarrow (a \mathbf{k} M \rightarrow \phi)$
7.  $\models (M :_a^{\mathcal{C}} \phi) \rightarrow \bigwedge_{b \in \mathcal{C} \cup \{a\}} (\llbracket M \rrbracket_a :_b^{\mathcal{C} \cup \{a\}} (a \mathbf{k} M \wedge M :_a^{\mathcal{C}} \phi))$
8.  $\models (M :_a^{\mathcal{C} \cup \mathcal{C}'} \phi) \rightarrow M :_a^{\mathcal{C}} \phi$
9. *If  $\models \phi$  then  $\models M :_a^{\mathcal{C}} \phi$*
10. *If  $\models a \mathbf{k} M \rightarrow a \mathbf{k} M'$  then  $\models (M' :_a^{\mathcal{C}} \phi) \rightarrow M :_a^{\mathcal{C}} \phi$ .*

*Proof.* 1–4 are immediate; 5, 7, and 10 follow directly from Proposition 5.2.b, 5.2.c, and 5.2.a, respectively; 6 follows directly from the reflexivity of  $\leq_{\mathcal{C} \cup \{a\}}$  and  $\equiv_a$ ; 8 follows directly from Proposition 4.2; and 9 is immediate.  $\square$

**Definition 6** (Semantic consequence and equivalence).

- The formula  $\phi' \in \mathcal{L}$  is a *semantic consequence* of  $\phi \in \mathcal{L}$ , written  $\phi \Rightarrow \phi'$ , :iff for all models  $(\mathfrak{S}, \mathcal{V})$  and states  $s \in \mathcal{S}$ , if  $(\mathfrak{S}, \mathcal{V}), s \models \phi$  then  $(\mathfrak{S}, \mathcal{V}), s \models \phi'$ .
- $\phi' \in \mathcal{L}$  is *semantically equivalent* to  $\phi \in \mathcal{L}$ , written  $\phi \Leftrightarrow \phi'$ , :iff  $\phi \Rightarrow \phi'$  and  $\phi' \Rightarrow \phi$ .

**Fact 1.**  $\models \phi \rightarrow \phi'$  if and only if  $\phi \Rightarrow \phi'$

*Proof.* By expansion of definitions.  $\square$

## 2.1 Epistemic explication

As announced, our interactive proofs have an *epistemic explication* in terms of the epistemic impact that they effectuate with their intended interpreting agents (i.e., the knowledge of their proof goals). To see this, consider that the elementary definition of proof accessibility on Page 19 can be transformed by applying elementary-logical rules so that

$$\begin{array}{l}
 (\mathfrak{S}, \mathcal{V}), s \models M \stackrel{\mathcal{C}}{:_a} \phi \quad \text{if and only if} \\
 \text{for all } \tilde{s} \in \mathcal{S}, \text{ if } s \leq_{\mathcal{C} \cup \{a\}} \tilde{s} \text{ then } \quad (\text{data } [\tilde{s}] \text{ and peer } [\mathcal{C} \cup \{a\}] \text{ persistent}) \\
 (\mathfrak{S}, \mathcal{V}), \tilde{s} \models a \underbrace{k M}_{\text{sufficient evidence}} \rightarrow K_a(\underbrace{\phi}_{\text{induced knowledge}}) \quad (\text{epistemic impact}),
 \end{array}$$

with the standard epistemic modality  $K_a$  being defined as

$$(\mathfrak{S}, \mathcal{V}), \tilde{s} \models K_a(\phi) \quad \text{:iff} \quad \text{for all } s' \in \mathcal{S}, \text{ if } \tilde{s} \equiv_a s' \text{ then } (\mathfrak{S}, \mathcal{V}), s' \models \phi.$$

As required,  $K_a$ —being defined by means of an equivalence relation—is S5, i.e., S4 plus the property  $\models \neg K_a(\phi) \rightarrow K_a(\neg K_a(\phi))$  of negative introspection [FHMV95, MV07]. Hence, spelled out, the epistemic explication is:

A proof effectuates a persistent epistemic impact in its intended community of peer reviewers that consists in the induction of the (propositional) knowledge of the proof goal by means of the (individual) knowledge of the proof with the interpreting reviewer.

Observe that our notion of *knowledge induction* (impact effectuation) is an instance of a *parameterised persistent implication*, which:

1. is compatible with C.I. Lewis relevant implication (a.k.a. *strict implication*), which does not stipulate any constraint on the accessibility relation of the implication (here  $\leq_{\mathcal{C} \cup \{a\}}$ )
2. is *intuitionistic implication* in Kripke's interpretation when the preorder  $\leq_{\mathcal{C} \cup \{a\}}$  happens to be partial, e.g., when  $\mathcal{A} = \{a\}$  (total knowledge).



D. Lewis relevant implication however (and *a fortiori* Stalnaker's) is insufficient for capturing the induction. Recall that a statement  $\phi$  implies  $\phi'$  in a state  $s$ , by definition of D. Lewis, if and only if  $\phi \rightarrow \phi'$  is true at all states closest to  $s$  (here with respect to  $\leq_{C \cup \{a\}}$ ). (Stalnaker required that there be a *single* closest state.) Order-theoretically, “closest to  $s$  with respect to  $\leq_{C \cup \{a\}}$ ” means “that are atomic (i.e., if minored then only by bottom) in the up-set  $\uparrow_{\leq_{C \cup \{a\}}}(s) := \{s' \in \mathcal{S} \mid s \leq_{C \cup \{a\}} s'\}$  of  $s$  with respect to  $\leq_{C \cup \{a\}}$ ”. Yet we do need to stipulate truth at all states *close* to  $s$  (i.e., *all* states in  $\uparrow_{\leq_{C \cup \{a\}}}(s)$ ), not just truth at all states *closest* (i.e., all *atomic* states). Otherwise persistency, which is essential to obtaining intuitionistic logic, may fail (cf. [vB97, Section 2] and [vB09]).

Still, we believe that D. Lewis relevant implication could be suitable for defining induction of *belief* (to be enshrined in a *Logic of Evidence*) and even *false belief* (to be enshrined in a *Logic of Deception*). For belief, it does not make sense to insist on (peer) persistency, except perhaps for *religious belief* (among sectarian peers), and so quantifying over all closest states could be preferable over quantifying over all close states. To be explored in future work.

We close this section with the statement of five epistemic interaction laws. The first law—to be used as a lemma for the second—describes a reflexive interaction between individual and propositional knowledge in the following sense.

**Proposition 7** (Self-knowledge).

$$\models K_a(a \mathbf{k} M) \leftrightarrow a \mathbf{k} M$$

*Proof.* The  $\rightarrow$ -direction follows from the reflexivity of  $\equiv_a$ , and the  $\leftarrow$ -direction from the definition of  $\equiv_a$  as state indistinguishability with respect to individual knowledge.  $\square$

The second law describes an important interaction between individual and propositional knowledge by means of their respective languages  $\mathcal{M}$  and  $\mathcal{L}$ . For the sake of stating the law succinctly, we recall the following standard definition.

**Definition 7** (Language equivalence). Let  $L \subseteq \mathcal{L}$  designate a sublanguage of  $\mathcal{L}$ . Then two pointed models  $(\mathfrak{S}, \mathcal{V}), s$  and  $(\mathfrak{S}, \mathcal{V}), s'$  are *L-equivalent*, written  $(\mathfrak{S}, \mathcal{V}), s \equiv_L (\mathfrak{S}, \mathcal{V}), s'$ , :iff for all  $\phi \in L$ ,  $(\mathfrak{S}, \mathcal{V}), s \models \phi$  iff  $(\mathfrak{S}, \mathcal{V}), s' \models \phi$ . (The relation  $\equiv_L$  is called *elementary equivalence*.)

The law says that state indistinguishability with respect to individual knowledge equals state indistinguishability with respect to propositional knowledge.

**Proposition 8** (Indistinguishability). *Let  $a \in \mathcal{A}$  and*

$$\begin{aligned} Re &:= \{ a \mathbf{k} M \mid M \in \mathcal{M} \} \\ Dicto &:= \{ K_a(\phi) \mid \phi \in \mathcal{L} \}. \end{aligned}$$

*Then,*

$$\equiv_{Re} = \equiv_{Dicto}.$$

*Proof.* The  $\subseteq$ -direction follows from the definition of  $\equiv_a$  as state indistinguishability with respect to individual knowledge, and the transitivity of  $\equiv_a$ ; and the  $\supseteq$ -direction from the fact that for all  $M \in \mathcal{M}$ ,  $(a \mathbf{k} M) \in \mathcal{L}$  and Proposition 7.  $\square$

The third law—to be used as a lemma for the fourth—describes an important interaction between individual and propositional knowledge by means of message signing. The law also gives an example of interpreted communication, i.e., how to induce propositional knowledge with a certain piece of individual knowledge (i.e., a signed message).

**Proposition 9** (The purpose of signing).

$$\models a \mathbf{k} \llbracket M \rrbracket_b \rightarrow K_a(b \mathbf{k} \llbracket M \rrbracket_b)$$

*Proof.* Let  $(\mathfrak{S}, \mathcal{V})$  designate an arbitrary LiP-model, and let  $s \in \mathcal{S}$ ,  $a, b \in \mathcal{A}$ , and  $M \in \mathcal{M}$ . Further, suppose that  $(\mathfrak{S}, \mathcal{V}), s \models a \mathbf{k} \llbracket M \rrbracket_b$  and let  $s' \in \mathcal{S}$  such that  $s' \equiv_a s$ . Hence,  $(\mathfrak{S}, \mathcal{V}), s' \models a \mathbf{k} \llbracket M \rrbracket_b$  by definition of  $\equiv_a$  as state indistinguishability with respect to individual knowledge, and thus  $(\mathfrak{S}, \mathcal{V}), s' \models b \mathbf{k} \llbracket M \rrbracket_b$  due to the unforgeability of signatures (only  $b$  can generate  $\llbracket M \rrbracket_b$ , cf. Page 11).  $\square$

The fourth law describes an important interaction between knowledge and interactive proofs, again by means of message signing. The law also gives an explication of the epistemic impact of *signed* interactive proofs.

**Theorem 3** (Proofs of Knowledge). *Signed interactive proofs are peer-reviewable proofs of knowledge<sup>12</sup> in the following formal sense:*

$$\models (M :_a^{\mathcal{C}} \phi) \rightarrow \bigwedge_{b \in \mathcal{C} \cup \{a\}} \llbracket M \rrbracket_a :_b^{\mathcal{C} \cup \{a\}} (\underbrace{a \mathbf{k} M \wedge K_a(\phi)}_{\text{induced knowledge}}).$$

*Proof.* We first prove the stronger fact that

$$\models (M :_a^{\mathcal{C}} \phi) \rightarrow \bigwedge_{b \in \mathcal{C} \cup \{a\}} \llbracket M \rrbracket_a :_b^{\mathcal{C} \cup \{a\}} (a \mathbf{k} M \wedge M :_a^{\mathcal{C}} \phi).$$

Let  $(\mathfrak{S}, \mathcal{V})$  designate an arbitrary LiP-model, and let  $s \in \mathcal{S}$ ,  $a \in \mathcal{A}$ ,  $\mathcal{C} \subseteq \mathcal{A}$ ,  $b \in \mathcal{C} \cup \{a\}$ , and  $M \in \mathcal{M}$ . Further, suppose that  $(\mathfrak{S}, \mathcal{V}), s \models M :_a^{\mathcal{C}} \phi$ , let  $\tilde{s} \in \mathcal{S}$  such that  $s \leq_{\mathcal{C} \cup \{a\} \cup \{b\}} \tilde{s}$ , and suppose that  $(\mathfrak{S}, \mathcal{V}), \tilde{s} \models b \mathbf{k} \llbracket M \rrbracket_a$ . Hence,  $(\mathfrak{S}, \mathcal{V}), \tilde{s} \models K_b(a \mathbf{k} \llbracket M \rrbracket_a)$  by Proposition 9, and thus  $(\mathfrak{S}, \mathcal{V}), \tilde{s} \models K_b(a \mathbf{k} M)$  by *modus ponens* of  $\models K_b(a \mathbf{k} \llbracket M \rrbracket_a \rightarrow a \mathbf{k} M)$  (epistemic necessitation of signature analysis) and  $\models K_b(a \mathbf{k} \llbracket M \rrbracket_a \rightarrow a \mathbf{k} M) \rightarrow (K_b(a \mathbf{k} \llbracket M \rrbracket_a) \rightarrow K_b(a \mathbf{k} M))$  (Kripke’s law). Now, let  $\tilde{s} \in \mathcal{S}$  such that  $\tilde{s} \equiv_b \tilde{s}$ . Thus,  $\tilde{s} \leq_{\mathcal{C} \cup \{a\} \cup \{b\}} \tilde{s}$ , thus  $s \leq_{\mathcal{C} \cup \{a\} \cup \{b\}} \tilde{s}$  by transitivity, and thus  $s \leq_{\mathcal{C} \cup \{a\}} \tilde{s}$  by the hypothesis that  $b \in \mathcal{C} \cup \{a\}$ . Hence,

<sup>12</sup>This terminology is inspired by [Gol01, Page 262], where such proofs are defined as “[...] proofs in which the prover [here  $a$ ] asserts “knowledge” of some object [...] and not merely its existence [...]” by means of probabilistic polynomial-time interactive Turing machines.

$(\mathfrak{S}, \mathcal{V}), \tilde{s} \models M :_a^{\mathcal{C}} \phi$  by peer persistency,  $(\mathfrak{S}, \mathcal{V}), \tilde{s} \models K_b(M :_a^{\mathcal{C}} \phi)$  by discharge of the last hypothesis, and thus  $(\mathfrak{S}, \mathcal{V}), \tilde{s} \models K_b(a \text{ k } M \wedge M :_a^{\mathcal{C}} \phi)$ .

Our theorem now follows from a stronger version of epistemic truthfulness, i.e.,  $\models (M :_a^{\mathcal{C}} \phi) \rightarrow (a \text{ k } M \rightarrow K_a(\phi))$ , which in turn follows from the expansion of the truth condition of  $M :_a^{\mathcal{C}} \phi$ .  $\square$

The fifth law describes an important interaction between *common knowledge* [FHMV95, MV07] and purported interactive proofs, namely their falsifiability in a *communal* sense of *Popper's critical rationalism*. More precisely, we refer to Popper's dictum that a hypothesis (here, that a purported interactive proof is indeed a proof) should be falsifiable in the sense that **if the hypothesis is false then its falsehood should be cognisable** (here, commonly knowable). In the present paper, we restrict the relation between Popper's *œuvre* and our work to this succinct dictum. Recall from [FHMV95, MV07] that common knowledge among a community  $\mathcal{C}$  can be captured with a modality  $CK_{\mathcal{C}}$  defined as

$$(\mathfrak{S}, \mathcal{V}), s \models CK_{\mathcal{C}}(\phi) \quad \text{iff} \quad \text{for all } s' \in \mathcal{S}, \text{ if } s \equiv_{\mathcal{C}} s' \text{ then } (\mathfrak{S}, \mathcal{V}), s' \models \phi,$$

where  $\equiv_{\mathcal{C}} := (\bigcup_{a \in \mathcal{C}} \equiv_a)^*$ . The intuition is that a statement  $\phi$  is common knowledge in a community  $\mathcal{C}$  of agents when: all agents know that  $\phi$  is true (call this new statement  $\phi'$ ), all agents know that  $\phi'$  is true (call this new statement  $\phi''$ ), all agents know that  $\phi''$  is true (call this new statement  $\phi'''$ ), etc. Note that depending on the properties of the employed communication lines, common knowledge may have to be pre-established off those lines along other lines [HM90].

**Theorem 4** (Falsifiability of interactive “proofs”). *Interactive “proofs” are falsifiable in a communal sense of Popper's, i.e., if a datum  $M \in \mathcal{M}$  is not a  $\mathcal{C} \cup \{a\}$ -reviewable proof of a statement  $\phi \in \mathcal{L}$  then this fact is communally cognisable as such by  $\mathcal{C} \cup \{a\}$  in terms of the common knowledge among  $\mathcal{C} \cup \{a\}$  of that fact. Formally,*

$$\models (\neg M :_a^{\mathcal{C}} \phi) \rightarrow CK_{\mathcal{C} \cup \{a\}}(\neg M :_a^{\mathcal{C}} \phi).$$

*Proof.* Let  $(\mathfrak{S}, \mathcal{V})$  designate an arbitrary LiP-model, and let  $s \in \mathcal{S}$ ,  $a \in \mathcal{A}$ ,  $\mathcal{C} \subseteq \mathcal{A}$ , and  $M \in \mathcal{M}$ . Further, suppose that  $(\mathfrak{S}, \mathcal{V}), s \models \neg M :_a^{\mathcal{C}} \phi$ , let  $s' \in \mathcal{S}$  such that  $s \equiv_{\mathcal{C} \cup \{a\}} s'$  (thus  $s' \leq_{\mathcal{C} \cup \{a\}} s$ ), and suppose by contradiction that  $(\mathfrak{S}, \mathcal{V}), s' \models M :_a^{\mathcal{C}} \phi$ . Hence  $(\mathfrak{S}, \mathcal{V}), s \models M :_a^{\mathcal{C}} \phi$  by peer persistency—contradiction!  $\square$

Note also the following simpler fact, which asserts that what is commonly accepted as proof constitutes common knowledge.

**Fact 2** (Common proof knowledge).

$$\models (M :_a^{\mathcal{C}} \phi) \rightarrow CK_{\mathcal{C} \cup \{a\}}(M :_a^{\mathcal{C}} \phi)$$

This however does *not* mean that  $M$  is known by everybody in  $\mathcal{C} \cup \{a\}$ !

## 2.2 Oracle-computational explication

As announced, our interactive proofs have also an *oracle-computational explication* in terms of a computation oracle that acts as a hypothetical provider and thus as an imaginary epistemic source of our interactive proofs. To see this, consider that the elementary definition of proof accessibility in Definition 3 can be *redefined* (for the time being) such that for all  $s, s' \in \mathcal{S}$ ,

$$\begin{aligned}
 s \text{ } {}_M\text{R}_a^C \text{ } s' &: \text{iff } s' \in \bigcup_{\substack{s <_{\mathcal{C} \cup \{a\}}^M \tilde{s} \text{ and} \\ M \in \text{cl}_a^{\tilde{s}}(\emptyset)}} [\tilde{s}]_{\equiv_a} \\
 \text{for all } M \in \mathcal{M} \text{ and } \mathcal{C} \subseteq \mathcal{A}, & <_{\mathcal{C}}^M := \left( \bigcup_{a \in \mathcal{C}} <_a^M \right)^{++} \\
 s <_a^M s' &: \text{iff } \text{cl}_a^s(\{M\}) = \text{cl}_a^{s'}(\emptyset),
 \end{aligned} \tag{2}$$

where ‘ $++$ ’ designates the closure operation of so-called *generalised transitivity* in the sense that  $<_{\mathcal{C}}^M \circ <_{\mathcal{C}}^{M'} \subseteq <_{\mathcal{C}}^{(M, M')}$ . Note that when  $s <_a^M s'$  for some states  $s, s' \in \mathcal{S}$ , agent  $a$  can conceive of  $s'$  as  $s$  yet *minimally enriched* with the information token  $M$ , for which  $a$  could imagine invoking an *oracle agent*. In other words, if  $a$  knew  $M$  (e.g., if  $a$  *received*  $M$  from the oracle) then  $a$  could not distinguish  $s$  from  $s'$  in the sense of  $\equiv_a$ . This hypothetical knowledge was called *adductive knowledge* in [Kra08b]—from now on also *oracle knowledge*—and implemented with a concrete message reception *event* for  $a$  that carries the information of  $M$  in  $s'$ . Now, similarly to Page 24, our above-redefined proof-accessibility relation can be transformed and then used for redefining (again, for the time being) the proof modality as follows:

$$\begin{aligned}
 (\mathfrak{S}, \mathcal{V}), s \models M :_a^C \phi &: \text{iff} \\
 \text{for all } s' \in \mathcal{S}, \text{ if } s <_{\mathcal{C} \cup \{a\}}^M s' &\text{ then } (\text{peer } [\mathcal{C} \cup \{a\}] \text{ persistent}) \\
 (\mathfrak{S}, \mathcal{V}), s' \models a \text{ k } M \rightarrow K_a(\phi) & \quad (\text{epistemic impact}).
 \end{aligned}$$

The new notion of proof resulting from Accessibility Relation 2 on Page 28 is obviously weaker than our original notion resulting from Accessibility Relation 1 on Page 19, in the sense that the epistemic impact of Notion 1 is *data persistent*, e.g., is the case even when more messages than just the proof are learnt, whereas the one of Notion 2 is not necessarily so, i.e., is the case possibly only at the instant of learning the proof. (Still, both notions induce knowledge and not only belief!) Therefore, we call interactive proofs in the sense of Notion 1 *persistent* or *extant* and those in the sense of Notion 2 *instant* interactive proofs. For multi-agent distributed systems, instant interactive proofs are interesting, e.g., for *accountability* (cf. [KR10] and [KR11], both based on [Kra08b]). In accountable multi-agent distributed systems, an agent may prove her correct past behaviour in the present state to some judge, e.g., with a signed logfile [KR10], but may well then cease behaving correctly in the future. Hence her correctness proof is instant but may well not be persistent. The epistemic explication for Notion 2 is, spelled out:

An *instant* proof effectuates an *instant* epistemic impact in its intended community of peer reviewers that consists in the induction of the (propositional) knowledge of the proof goal by means of the (individual) knowledge of the proof with the interpreting reviewer.

Observe that our notion of knowledge induction (impact effectuation) for instant interactive proofs is a parameterised instant implication, which *is* compatible with D. Lewis relevant implication (cf. our corresponding discussion on Page 25). That is,  $a \mathbf{k} M \rightarrow K_a(\phi)$  is true at all states  $s'$  closest to  $s$  with respect to  $\leq_{C \cup \{a\}}$ , i.e., for which  $s <_{C \cup \{a\}}^M s'$ . The token  $M$  represents the minimal difference. Of course,  $a$  may in fact know  $M$  in  $s$ ; so the conditional is *not necessarily counterfactual*.

Our above definitions can be related to our original ones as follows.

**Proposition 10.** *For all  $s, s' \in \mathcal{S}$ :*

1.  $s \leq_a s'$  if and only if there is  $M \in \mathcal{M}$  such that  $s <_a^M s'$
2.  $s \leq_C s'$  if and only if there is  $M \in \mathcal{M}$  such that  $s <_C^M s'$

*Proof.* We prove the if-direction of (1)—the only-if-direction being obvious, and (2) obviously following from (1). Let  $s, s' \in \mathcal{S}$  and suppose that  $s \leq_a s'$ . Hence there is a finite  $\mathcal{D} \subseteq \mathcal{M}$  such that  $\text{cl}_a^s(\mathcal{D}) = \text{cl}_a^{s'}(\emptyset)$ , because  $\text{msgs}_a(s)$  and  $\text{msgs}_a(s')$  are finite (cf. Page 18). Hence there is  $M \in \mathcal{M}$  such that  $\text{cl}_a^s(\{M\}) = \text{cl}_a^s(\mathcal{D})$ . Thus,  $\text{cl}_a^s(\{M\}) = \text{cl}_a^{s'}(\emptyset)$  by transitivity, and  $s <_a^M s'$  by definition.  $\square$

Hence, Notion 1 can be recovered from Notion 2 by *redefining* the proof accessibility on Page 19 such that for all  $s, s' \in \mathcal{S}$ ,

$$s \mathbf{M}R_a^C s' \quad \text{iff} \quad s' \in \bigcup [\tilde{s}]_{\equiv_a}, \quad (3)$$

$$s (\bigcup_{M' \in \mathcal{M}} <_{C \cup \{a\}}^{M'}) \tilde{s} \quad \text{and} \quad M \in \text{cl}_a^s(\emptyset)$$

and thus Notion 3 and Notion 1 are equivalent.

**Proposition 11.** *When the proof modality is interpreted with Notion 2,*

$$\models a :_a^\emptyset \phi \leftrightarrow K_a(\phi).$$

*Proof.* By the fact that  $<_{\{a\}}^a = \equiv_a$ .  $\square$

We leave the further study of instant interactive proofs for future work.

## 2.3 More results

**Theorem 5** (Adequacy).  $\vdash_{\text{LiP}}$  is adequate for  $\models$ , i.e.,:

1. if  $\vdash_{\text{LiP}} \phi$  then  $\models \phi$  (axiomatic soundness)
2. if  $\models \phi$  then  $\vdash_{\text{LiP}} \phi$  (semantic completeness).

*Proof.* Soundness follows from the admissibility of axioms and rules (cf. Proposition 6), and completeness by means of the classical construction of canonical models, using Lindenbaum's construction of maximally consistent sets (cf. Appendix A).  $\square$

We leave the study of *strong adequacy* [Fit07, Section 3] for future work.

**Corollary 3** (Consistency).

1. If  $\vdash_{\text{LiP}} \phi$  then  $\not\vdash_{\text{LiP}} \neg\phi$ .
2.  $\not\vdash_{\text{LiP}} \perp$

*Proof.* As usual: suppose that  $\vdash_{\text{LiP}} \phi$ . Hence  $\models \phi$  by semantic completeness. Hence  $\not\models \neg\phi$  by definition of  $\models$ . Hence  $\not\vdash_{\text{LiP}} \neg\phi$  by contraposition of axiomatic soundness; and (2) follows jointly from the instance of (1) where  $\phi := \top$ , the axiom  $\vdash_{\text{LiP}} a \text{ k } a$ , and the macro-definitions of  $\top$  as  $a \text{ k } a$  and  $\perp$  as  $\neg\top$ .  $\square$

However *negation completeness* (i.e., “ $\vdash_{\text{LiP}} \phi$  or  $\vdash_{\text{LiP}} \neg\phi$ ”) fails for LiP, as for classical propositional logic, which is a fragment of LiP. As a consequence, LiP does not have the *disjunction property* (i.e., “if  $\vdash_{\text{LiP}} \phi \vee \phi'$  then  $\vdash_{\text{LiP}} \phi$  or  $\vdash_{\text{LiP}} \phi'$ ”); for example consider the case where  $\phi' := \neg\phi$ .

**Corollary 4** (Stateful proof equality). *Let  $(\mathfrak{S}, \mathcal{V})$  designate an arbitrary LiP-model, and let  $s \in \mathcal{S}$ ,  $M \in \mathcal{M}$ ,  $a \in \mathcal{A}$ ,  $\mathcal{C} \subseteq \mathcal{A}$ , and  $\phi \in \mathcal{L}$ . Further let:*

$$\begin{aligned} s \equiv_a^{\mathcal{C}} &:= \{ (M, M') \in \mathcal{M} \times \mathcal{M} \mid s \llbracket M \rrbracket_a^{\mathcal{C}} = s \llbracket M' \rrbracket_a^{\mathcal{C}} \} \\ 0 &:= [a]_{s \equiv_a^{\mathcal{C}}} \\ [M]_{s \equiv_a^{\mathcal{C}}} + [M']_{s \equiv_a^{\mathcal{C}}} &:= [(M, M')]_{s \equiv_a^{\mathcal{C}}}. \end{aligned}$$

Then,

$$\langle \mathcal{M} /_{s \equiv_a^{\mathcal{C}}}, 0, + \rangle$$

is an idempotent commutative monoid, i.e., for all  $\mathbf{M}, \mathbf{M}', \mathbf{M}'' \in \mathcal{M} /_{s \equiv_a^{\mathcal{C}}}$ :

1.  $\mathbf{M} + (\mathbf{M}' + \mathbf{M}'') = (\mathbf{M} + \mathbf{M}') + \mathbf{M}''$  (associativity)
2.  $\mathbf{M} + \mathbf{M}' = \mathbf{M}' + \mathbf{M}$  (commutativity)
3.  $\mathbf{M} + \mathbf{M} = \mathbf{M}$  (idempotency)
4.  $\mathbf{M} + 0 = \mathbf{M}$  (neutral element).

*Proof.* By the soundness of proof associativity, commutativity, and idempotency, and the law of a self-neutral proof element, respectively (cf. Theorem 1).  $\square$

### 3 Related work

In this section, we relate our Logic of interactive Proofs (LiP) to Artëmov’s Logic of Proofs (LP) [Art94] and to a generalised variant thereof, namely his Symmetric Logic of Proofs (SLP) [Art08b]. We also relate LiP to two extensions of LP with multi-agent character, namely Yavorskaya’s LP<sup>2</sup> [Yav08] and Renne’s UL [Ren12]. The general aim of this section is to give a detailed description of crucial design decisions for interactive and non-interactive systems on the example of related works. Essentially, we argue that, first, LP and LiP can be related but have typically different (not always) but complementary scopes, namely non-interactive computation and universal truths, and interactive computation and local truths, respectively; and, second, LiP improves LP-like systems with respect to interactivity. That LP and LiP can indeed be related is evidenced to some extent by Theorem 6 and proved by the example following it, which happens to be formalisable in both LP and LiP. That we discuss multi-agent extensions of LP is justified by the fact that LP<sup>2</sup> and UL are intended to be interactive but inherit the lack of message-passing interactivity from LP. As a matter of fact, the example with *signing* is formalisable only in LiP.

#### 3.1 Concepts

In (S)LP,  $p:F$  stands for an atomic concept. Whereas in LiP,  $M :_a^C \phi$  stands for a compound concept analysable into epistemic constituents (cf. Section 2.1), *nota bene* thanks to a constructive semantics defined in terms of the proof terms themselves (cf. Page 19). In that, our construction is reminiscent of the canonical-model construction, which like ours is a constructive semantics defined in terms of syntax, but unlike ours not in terms of terms but in terms of formulas (cf. Appendix A).

##### 3.1.1 Interactivity

(S)LP proofs are non-interactive, whereas LiP proofs are interactive (knowledge-inducing). (S)LP proofs are non-interactive also due to (S)LP’s reflection axiom, which stipulates that provability imply truth<sup>13</sup>. However, in a truly interactive setting, (S)LP’s reflection axiom is unsound. By a truly interactive setting, we mean a multi-agent distributed system where not all proofs are known by all agents, i.e., a setting with a non-trivial distribution of information in the sense of Scott (cf. Proposition 3), in which  $\not\models a \text{ k } M$ . In other words, in truly interactive settings, agents are not omniscient with respect to messages. Otherwise, why communicate? As proof, consider the following, self-referential counter-example:  $\models M :_a^\emptyset (a \text{ k } M)$  (self-knowledge) but  $\not\models M :_a^\emptyset (a \text{ k } M) \rightarrow a \text{ k } M$ . In truly interactive settings, there being a proof does not imply knowledge of that proof. When an agent  $a$  does not know the proof and the agent cannot generate the proof *ex nihilo* herself by guessing it, only *communication* from a peer, who thus acts as an oracle, can entail the knowledge of the proof with  $a$ . In sum, **provability**

<sup>13</sup>(S)LP (and LiP) has a semantics, so we may use the word ‘truth’ here.

*and truth are necessarily concomitant in the non-interactive setting, whereas in interactive settings they are not necessarily so.*

### 3.1.2 Proof terms

(S)LP needs three proof-term constructors, namely sum, application, and proof checker. Whereas LiP only needs two, namely pairing and signing. Incidentally, Gödel conjectured that two proof-term constructors were sufficient for proofs [Art01]. In LiP, pairing plays a pair of roles, namely the two roles played by sum and application in LP, and thanks to Fact 2 the agents themselves within their own communities may—not a term constructor like ‘!’ in (S)LP must—play the proof-checker role! In sum, first, LiP-agents play a pair of roles, namely the two roles of proof as well as signature checker, and, second, signatures can be conceived as proof-checker-apposed, communally verifiable seals of check.

### 3.1.3 Formulas

(S)LP’s proof modality ‘:’ has no parameters, whereas LiP’s ‘:<sub>a</sub><sup>C</sup>’ has two. The advantage of LiP’s parametric modality is agent-centricity and thus greater generality. As a nice side effect, LiP’s proof terms have neutral elements.

## 3.2 Laws

### 3.2.1 Structural laws (cf. Theorem 1)

In LP, the proof-sum operation ‘+’ is neither commutative nor idempotent, but in SLP, it is both, like ‘(·, ·)’ in LiP. In (S)LP, ‘+’ has no neutral element, whereas in LiP the corresponding ‘(·, ·)’ has. As said previously, LiP’s ‘(·, ·)’ can simulate not only LP’s proof sum but also (S)LP’s proof application. However, LiP’s ‘(·, ·)’ *cannot* simulate SLP’s sum. To see why, consider that if (S)LP were defined analogously to LiP by means of a separate term theory using atomic propositions ‘kp’ (for “p is known”) and an analog of epistemic antitonicity then the structural modal laws of (S)LP could be (partially) generated from the structural term laws, analogously to LiP.

**LP** From the term axiom schema

$$kp+q \rightarrow (kp \wedge kq)$$

generate the corresponding characteristic law

$$((p:F) \vee q:F) \rightarrow (p+q):F.$$

**SLP**

1. From the term axiom schema

$$kp+q \rightarrow (kp \wedge kq)$$



generate the corresponding characteristic law

$$((p:F) \vee q:F) \rightarrow (p+q):F.$$

2. Add the axiom schema

$$((p+q):F) \rightarrow ((p:F) \vee q:F),$$

and (disregarding SLP's proof application) obtain the characteristic law

$$((p:F) \vee q:F) \leftrightarrow (p+q):F$$

of SLP's sum, which subsumes LP's sum law.

However in the case of LiP,

$$\not\models ((M, M') :_a^C \phi) \rightarrow ((M :_a^C \phi) \vee M' :_a^C \phi),$$

due to the obvious counter-example (recall that  $\models (M, M') :_a^\emptyset a \mathbf{k} (M, M')$ )

$$\not\models ((M, M') :_a^\emptyset a \mathbf{k} (M, M')) \rightarrow ((M :_a^\emptyset a \mathbf{k} (M, M')) \vee M' :_a^\emptyset a \mathbf{k} (M, M')).$$

That is, it is not generally true that single projections prove pair knowledge.

### 3.2.2 Logical laws (cf. Theorem 2)

(S)LP does not obey Kripke's law K, the law of necessitation, nor a law of modal idempotency. Whereas LiP does obey K as well as the generalised Kripke-law GK, necessitation, and the law of modal idempotency.

Observe that K is deducible from GK in LiP due to proof idempotency, which in turn is deducible in LiP due to pairing idempotency, which in turn is deducible in LiP due to conjunction idempotency and the pairing axiom (cf. Section B.1). Note that for resource-bounded agents, restricting the (resource-unbounded) pairing axiom would be desirable in order to prevent the (resource-unbounded) K from being deducible in LiP. Incidentally, (S)LP can be understood as being reconstructed only from the (resource-bounded) *unpairing* axiom and not from the (resource-unbounded) pairing axiom (cf. Section 3.2.1).

The justification for choosing (plain) necessitation instead of LP's constant specification for LiP is that in the interactive setting, validities, and thus *a fortiori* tautologies (in the strict sense of validities of the propositional fragment), are in some sense trivialities. To see why, recall from Definition 5 that validities are true in *all* pointed models, and thus not worth being communicated from one point to another in a given model, e.g., by means of specific interactive proofs. (Nothing is logically more embarrassing than talking in tautologies.) Therefore, validities deserve *arbitrary* messages as proof. What is worth being communicated are truths weaker than validities, namely local truths in the sense of Definition 5, which do not hold universally (cf. Table 2). Note that our choice is not forced but free: we could have chosen constant specification for LiP too

Table 2: Interesting truths

Computation	Truth
interactive	local
non-interactive	universal

(e.g., “ $\vdash_{\text{LiP}} a :_a^C \phi$ , for  $\phi \in \Gamma_1$ ”) and thus kept a closer relationship between LP and LiP, but that would have, first, put unnecessarily strong proof obligations on validities as far as interactivity is concerned, as explained; and, second, unfaithfully modelled resource-unbounded interacting agents, which already know all universal truths or validities, though of course not all local truths, which is the whole point of interacting with each other!

(S)LP does not obey the law of modal idempotency, because it does not have agents that could act as proof checkers and thus needs a term constructor for proof-checking. Whereas LiP does obey modal idempotency, because LiP does have agents that can act as proof checkers (cf. Section 3.1.2) and thus does not need a term constructor for proof-checking. Observe that modal idempotency is deducible in LiP due to the law of self-signing elimination, which in turn is deducible in LiP due to the axiom of personal signature synthesis (cf. Section B.2). Note that for resource-bounded agents, restricting (resource-unbounded) personal signature synthesis could be desirable in order to prevent (resource-unbounded) modal idempotency from being deducible in LiP. Incidentally, (S)LP can be understood as being reconstructed from no term axioms involving the proof checker ‘!’ (cf. Section 3.2.1).

### 3.2.3 Meta-logical properties

(S)LP is not a normal modal logic, because (S)LP does not obey Kripke’s law. Whereas LiP is a normal logic (cf. Corollary 2). LP is in  $\Sigma_2^P$  [Kuz00], but the decidability and thus complexity of SLP is unknown [Art08b]. A lower complexity bound for LiP is EXPTIME, which follows from the complexity of the logic of common knowledge, which is EXPTIMEcomplete [HM92], and from the fact that the concrete accessibility relation  $_M R_a^C$  for LiP requires  $\leq_{C \cup \{a\}}$ , which contains the one for common knowledge  $\equiv_{C \cup \{a\}}$  (cf. Page 27).

### 3.3 Formal relation

In order to establish a formal relation between LP and LiP, we consider LiP over a singleton society and over the term forms suggested on Page 13. So without loss of generality let  $\mathcal{A} = \{a\}$ . Further, fix LP’s set of specification constants to consist of  $\{a\}$ , and consider the mapping  $h$  over LP-formulas that maps LP’s

- proof-sum ‘+’ and proof-application ‘ $\cdot$ ’ to LiP’s proof-pair constructor ‘ $(\cdot, \cdot)$ ’
- proof checker ‘!’ to LiP’s proof-signature constructor ‘ $\{\!\cdot\!\}_a$ ’

- proof modality ‘ $\cdot$ ’ to LiP’s proof modality ‘ $\cdot_a^\emptyset$ ’.

**Lemma 1** (Admissibility of LP-laws for LiP). *When  $\mathcal{A} = \{a\}$ :*

0.  $\vdash_{\text{LiP}} \varphi$ , for any axiom  $\varphi$  of classical propositional logic
1.  $\vdash_{\text{LiP}} ((M :_a^\emptyset \phi) \vee M' :_a^\emptyset \phi) \rightarrow (M, M') :_a^\emptyset \phi$
2.  $\vdash_{\text{LiP}} (M :_a^\emptyset (\phi \rightarrow \phi')) \rightarrow ((M' :_a^\emptyset \phi) \rightarrow (M, M') :_a^\emptyset \phi')$
3.  $\vdash_{\text{LiP}} (M :_a^\emptyset \phi) \rightarrow \phi$
4.  $\vdash_{\text{LiP}} (M :_a^\emptyset \phi) \rightarrow \llbracket M \rrbracket_a :_a^\emptyset (M :_a^\emptyset \phi)$
5.  $\{\phi \rightarrow \phi', \phi'\} \vdash_{\text{LiP}} \phi'$
6.  $\vdash_{\text{LiP}} a :_a^\emptyset \phi$ , for any formula  $\phi$  for which  $\vdash_{\text{LiP}} \phi$  in Item 0–4.

*Proof.* 0 holds by definition of LiP. For the rest, set  $\mathcal{C} = \emptyset$ . Then 1 is LiP’s law of proof extension (cf. Theorem 1.11); 2 is LiP’s axiom schema GK; 3 is, given that  $\mathcal{A} = \{a\}$ , LiP’s law of truthfulness (cf. Theorem 2.26.b); 4 follows directly from LiP’s laws of self-signing idempotency and modal idempotency (cf. Theorem 1.19 and 2.25); 5 holds by definition of LiP; and 6 follows by particularising LiP-necessitation.  $\square$

**Theorem 6** (Homomorphism from LP into LiP). *For  $\mathcal{A}$  a singleton, for all LP-formulas  $F$ ,*

$$\text{if } \vdash_{\text{LP}} F \text{ then } \vdash_{\text{LiP}} h(F).$$

*Proof.* From the admissibility of LP-axioms and LP-rules for LiP (cf. Lemma 1); or, by comparison of LP’s semantics with LiP’s semantics for singleton societies.  $\square$

However the converse is not true, and thus  $h$  is only a homomorphism and not an embedding. As counter example consider Kripke’s law, which holds in LiP (cf. Theorem 2.1), but does not hold in LP (cf. Section 3.2.2). In sum, while plain propositional logic can be viewed as a modal logic interpreted over a singleton universe, LP can be viewed only to a limited extent as LiP over a singleton society. The extent is limited because LiP does not mathematically contain LP, as LP does not embed (injectively homomorph) but only non-injectively homomorph into LiP, which we believe reflects the essential difference between their scopes. We stress that LP and LiP have typically different, complementary scopes, namely non-interactive computation and universal truths, and interactive computation and local truths, respectively. Nevertheless:

1. LP and LiP have a non-empty intersection, as the following example proves, which happens to be formalisable in both LP [Art08a] and LiP.
2. LiP is richer than S4, since LiP generalises S4 with agent centricity and refines S4 with explicit, transmittable proofs.

The example involves two elementary formal proofs, which for clarity we present in the style of Frederic Fitch, justified by the following definition and facts.

**Definition 8** (Local hypotheses). Let  $\Lambda \subseteq \mathcal{L}$  such that  $\Lambda$  is finite, and

$$\Gamma; \Lambda \vdash_{\text{LiP}} \phi \quad \text{iff} \quad \Gamma \vdash_{\text{LiP}} (\bigwedge \Lambda) \rightarrow \phi,$$

cf. Proposition 1, where  $\Lambda$  is understood as a finite set of local hypotheses.

**Fact 3** (*A fortiori* true, persistently provable and known as true).

$$\Gamma, \phi; \Lambda \vdash_{\text{LiP}} \phi \wedge (M :_a^C \phi) \wedge a :_a^\emptyset \phi,$$

where  $\Gamma, \phi$  means  $\Gamma \cup \{\phi\}$ .

*Proof.* From Proposition 1 by the fact that  $\vdash_{\text{LiP}} \phi$  implies  $\vdash_{\text{LiP}} \phi$ , necessitation, and self-truthfulness *bis* for the above case  $\phi$ ,  $M :_a^C \phi$ , and  $a :_a^\emptyset \phi$ , respectively.  $\square$

Recall from Section 2.1, that  $\vdash_{\text{LiP}} a :_a^\emptyset \phi$  can be read as “*a* persistently knows that  $\phi$  is true” (unless interpreted defeasibly, cf. Proposition 11).

**Fact 4** (Fitting-style deduction “theorems” [Fit07]).

$$\text{LDT} \frac{\Gamma; \Lambda, \phi \vdash_{\text{LiP}} \phi'}{\Gamma; \Lambda \vdash_{\text{LiP}} \phi \rightarrow \phi'} \quad \text{MP} \frac{\Gamma; \Lambda, \phi \vdash_{\text{LiP}} \phi'}{\Gamma, \phi; \Lambda \vdash_{\text{LiP}} \phi'}$$

Here, “LDT” abbreviates “Local Deduction Theorem”, “MP” abbreviates “modus ponens”,  $\Lambda, \phi$  means  $\Lambda \cup \{\phi\}$ , the double horizontal bar means “if and only if”, and the simple horizontal bar reads “if ... then ...” from top to bottom.

*Proof.* The validity of the LDT rule schema is warranted by Definition 8, and the one of the MP rule schema by the *modus ponens* rule schema of LiP.  $\square$

Following [Art08a], we now present the more difficult Case I of Gettier’s Case I and II, which according to [Art08a] “were supposed to provide examples of justified true beliefs which should not be considered knowledge.”

**Example** (Gettier, from [Art08a]). Suppose that Smith and Jones have applied for a certain job. And suppose that Smith has strong evidence for the following conjunctive proposition: (d) Jones is the man who will get the job, and Jones has ten coins in his pocket. Proposition (d) entails: (e) The man who will get the job has ten coins in his pocket. Let us suppose that Smith sees the entailment from (d) to (e), and accepts (e) on the grounds of (d), for which he has strong evidence. In this case, Smith is clearly justified in believing that (e) is true. But imagine, further, that unknown to Smith, he himself, not Jones, will get the job. And also, unknown to Smith, he himself has ten coins in his pocket. Then, all of the following are true: 1) (e) is true, 2) Smith believes that (e) is true. But it is equally clear that Smith does not know that (e) is true.

Interpreting “strong evidence” in Gettier’s example as “proof” in our sense, Gettier’s Case I can be formalised in LiP as follows. Let:

- $a \in \mathcal{A} := \{\text{Smith}, \text{Jones}\};$
- for all  $a \in \mathcal{A}$ ,  $\text{job}(a), 10(a) \in \mathcal{P}$ .

Then Gettier's assumptions stated in his example are contradictory, as asserted by Proposition 12 and proved by jointly Lemma 2 and the proof in Table 3. Lemma 2 corresponds to the assertion that (d) entails (e) in Gettier's example.

**Lemma 2** (Gettier example).

$$\{(\text{job}(\text{Smith}) \wedge \text{job}(\text{Jones})) \rightarrow \perp\}; \emptyset \vdash_{\text{LiP}} (\text{job}(\text{Jones}) \wedge 10(\text{Jones})) \rightarrow \bigwedge_{a \in \mathcal{A}} (\text{job}(a) \rightarrow 10(a))$$

*Proof.*

1.  $\vdash_{\text{LiP}} (\text{job}(\text{Smith}) \wedge \text{job}(\text{Jones})) \rightarrow \perp$  global hypothesis
2. (a)  $\text{job}(\text{Jones}) \wedge 10(\text{Jones})$  local hypothesis  
     (b)  $\text{job}(\text{Jones}) \rightarrow 10(\text{Jones})$  2.a, PL  
     (c)  $(\text{job}(\text{Smith}) \wedge \text{job}(\text{Jones})) \rightarrow \perp$  1, *a fortiori*  
     (d)  $\neg \text{job}(\text{Smith})$  2.a, 2.c, PL  
     (e)  $\text{job}(\text{Smith}) \rightarrow 10(\text{Smith})$  2.d, PL  
     (f)  $(\text{job}(\text{Jones}) \rightarrow 10(\text{Jones})) \wedge (\text{job}(\text{Smith}) \rightarrow 10(\text{Smith}))$  2.b, 2.e, PL  

$\underbrace{\hspace{15em}}$   
 the man who will get the job has 10 coins in his pocket
3.  $\vdash_{\text{LiP}} (\text{job}(\text{Jones}) \wedge 10(\text{Jones})) \rightarrow \bigwedge_{a \in \mathcal{A}} (\text{job}(a) \rightarrow 10(a))$  2.a–2.f, LDT
4. if  $\vdash_{\text{LiP}} (\text{job}(\text{Smith}) \wedge \text{job}(\text{Jones})) \rightarrow \perp$  1–3, PL  
     then  $\vdash_{\text{LiP}} (\text{job}(\text{Jones}) \wedge 10(\text{Jones})) \rightarrow \bigwedge_{a \in \mathcal{A}} (\text{job}(a) \rightarrow 10(a))$
5.  $\{(\text{job}(\text{Smith}) \wedge \text{job}(\text{Jones})) \rightarrow \perp\}; \emptyset \vdash_{\text{LiP}} (\text{job}(\text{Jones}) \wedge 10(\text{Jones})) \rightarrow \bigwedge_{a \in \mathcal{A}} (\text{job}(a) \rightarrow 10(a))$  4, definition.

□

**Proposition 12** (Gettier example).

$$\{(\text{job}(\text{Smith}) \wedge \text{job}(\text{Jones})) \rightarrow \perp\}; \emptyset \vdash_{\text{LiP}} (\text{Smith} \text{ k } M \wedge M :_{\text{Smith}}^{\emptyset} (\text{job}(\text{Jones}) \wedge 10(\text{Jones}))) \rightarrow ((\text{job}(\text{Smith}) \wedge 10(\text{Smith})) \rightarrow \perp)$$

*Proof.* See Table 3.

□

In order to illustrate the working of signatures and the application of the other logical laws of LiP, we now refine Gettier's example with signing. That is, we identify the proof  $M$  in Proposition 12 with a term pair  $(C, \llbracket W \rrbracket_{\text{HR}})$  consisting of, first, a proof  $C$  for the fact  $10(\text{Jones})$  and, second, a work contract  $\llbracket W \rrbracket_{\text{HR}}$  for  $\text{Jones}$  signed by the HR department dealing with the job application.

Table 3: Gettier example (proof of Proposition 12)

1.	$\vdash_{\text{LiP}} (\text{job}(\text{Smith}) \wedge \text{job}(\text{Jones})) \rightarrow \perp$	global hypothesis
2.	$\vdash_{\text{LiP}} (\text{job}(\text{Jones}) \wedge 10(\text{Jones})) \rightarrow \bigwedge_{a \in \mathcal{A}} (\text{job}(a) \rightarrow 10(a))$	1, Lemma 2, PL
3.	$\vdash_{\text{LiP}} \text{Smith} \cdot^{\emptyset}_{\text{Smith}} ((\text{job}(\text{Jones}) \wedge 10(\text{Jones})) \rightarrow \bigwedge_{a \in \mathcal{A}} (\text{job}(a) \rightarrow 10(a)))$	2, necessitation
4.	$\vdash_{\text{LiP}} \text{Smith} \text{ k Smith}$	knowledge of own's own name string
5.	(a) $\text{Smith} \text{ k } M \wedge M \cdot^{\emptyset}_{\text{Smith}} (\text{job}(\text{Jones}) \wedge 10(\text{Jones}))$ (b) $M \cdot^{\emptyset}_{\text{Smith}} (\text{job}(\text{Jones}) \wedge 10(\text{Jones}))$ (c) $\text{Smith} \cdot^{\emptyset}_{\text{Smith}} ((\text{job}(\text{Jones}) \wedge 10(\text{Jones})) \rightarrow \bigwedge_{a \in \mathcal{A}} (\text{job}(a) \rightarrow 10(a)))$ (d) $(\text{Smith}, M) \cdot^{\emptyset}_{\text{Smith}} (\bigwedge_{a \in \mathcal{A}} (\text{job}(a) \rightarrow 10(a)))$ (e) $\text{Smith} \text{ k Smith}$ (f) $\text{Smith} \text{ k } M$ (g) $\text{Smith} \text{ k Smith} \wedge \text{Smith} \text{ k } M$ (h) $\text{Smith} \text{ k } (\text{Smith}, M)$ (i) $\bigwedge_{a \in \mathcal{A}} (\text{job}(a) \rightarrow 10(a))$ (j) i. $\text{job}(\text{Smith}) \wedge 10(\text{Smith})$ ii. $\text{job}(\text{Smith})$ iii. $\text{Smith} \text{ k } M \wedge M \cdot^{\emptyset}_{\text{Smith}} (\text{job}(\text{Jones}) \wedge 10(\text{Jones}))$ iv. $\text{job}(\text{Jones}) \wedge 10(\text{Jones})$ v. $\text{job}(\text{Jones})$ vi. $\text{job}(\text{Smith}) \wedge \text{job}(\text{Jones})$ vii. $(\text{job}(\text{Smith}) \wedge \text{job}(\text{Jones})) \rightarrow \perp$ viii. $\perp$ (k) $(\text{job}(\text{Smith}) \wedge 10(\text{Smith})) \rightarrow \perp$	local hypothesis 5.a, PL 3, <i>a fortiori</i> 5.b, 5.c, GK 4, <i>a fortiori</i> 5.a, PL 5.e, 5.f, PL 5.g, pairing 5.d, 5.h, epistemic truthfulness local hypothesis 5.j.i, PL 5.a, <i>a fortiori</i> 5.j.iii, epistemic truthfulness 5.j.iv, PL 5.j.ii, 5.j.v, PL 1, <i>a fortiori</i> 5.j.vi, 5.j.vii, PL 5.j.i-5.j.viii, LDT 5.a-5.k, LDT
6.	$\vdash_{\text{LiP}} (\text{Smith} \text{ k } M \wedge M \cdot^{\emptyset}_{\text{Smith}} (\text{job}(\text{Jones}) \wedge 10(\text{Jones}))) \rightarrow ((\text{job}(\text{Smith}) \wedge 10(\text{Smith})) \rightarrow \perp)$	
7.	if $\vdash_{\text{LiP}} (\text{job}(\text{Smith}) \wedge \text{job}(\text{Jones})) \rightarrow \perp$ then $\vdash_{\text{LiP}} (\text{Smith} \text{ k } M \wedge M \cdot^{\emptyset}_{\text{Smith}} (\text{job}(\text{Jones}) \wedge 10(\text{Jones}))) \rightarrow ((\text{job}(\text{Smith}) \wedge 10(\text{Smith})) \rightarrow \perp)$	1-6, PL
8.	$\{(\text{job}(\text{Smith}) \wedge \text{job}(\text{Jones})) \rightarrow \perp\}; \emptyset \vdash_{\text{LiP}} (\text{Smith} \text{ k } M \wedge M \cdot^{\emptyset}_{\text{Smith}} (\text{job}(\text{Jones}) \wedge 10(\text{Jones}))) \rightarrow ((\text{job}(\text{Smith}) \wedge 10(\text{Smith})) \rightarrow \perp)$	7, definition.

**Lemma 3** (Gettier-example with signing). *Given  $\mathcal{A} := \{\text{Smith}, \text{Jones}, \text{HR}\}$ ,*

$$\vdash_{\text{LiP}} \left( \begin{array}{l} \text{Smith } k \{W\}_{\text{HR}} \wedge W :_{\text{HR}}^A \text{job}(\text{Jones}) \\ \wedge \text{Smith } k C \wedge C :_{\text{Smith}}^{\emptyset} 10(\text{Jones}) \end{array} \right) \rightarrow \\ \left( \begin{array}{l} \text{Smith } k (C, \{W\}_{\text{HR}}) \wedge \\ (C, \{W\}_{\text{HR}}) :_{\text{Smith}}^{\emptyset} (\text{job}(\text{Jones}) \wedge 10(\text{Jones})) \end{array} \right)$$

*Proof.*

1.  $\text{Smith } k \{W\}_{\text{HR}}$  local hypothesis
2.  $W :_{\text{HR}}^A \text{job}(\text{Jones})$  local hypothesis
3.  $(W :_{\text{HR}}^A \text{job}(\text{Jones})) \rightarrow \{W\}_{\text{HR}} :_{\text{Smith}}^A \text{job}(\text{Jones})$  simple peer review
4.  $\{W\}_{\text{HR}} :_{\text{Smith}}^A \text{job}(\text{Jones})$  2, 3, PL
5.  $(\{W\}_{\text{HR}} :_{\text{Smith}}^A \text{job}(\text{Jones})) \rightarrow \{W\}_{\text{HR}} :_{\text{Smith}}^{\emptyset} \text{job}(\text{Jones})$  group decomp.
6.  $\{W\}_{\text{HR}} :_{\text{Smith}}^{\emptyset} \text{job}(\text{Jones})$  4, 5, PL
7.  $\text{Smith } k C$  local hypothesis
8.  $C :_{\text{Smith}}^{\emptyset} 10(\text{Jones})$  local hypothesis
9.  $(C :_{\text{Smith}}^{\emptyset} 10(\text{Jones})) \rightarrow (C, \{W\}_{\text{HR}}) :_{\text{Smith}}^{\emptyset} 10(\text{Jones})$  proof ext.
10.  $(C, \{W\}_{\text{HR}}) :_{\text{Smith}}^{\emptyset} 10(\text{Jones})$  8, 9, PL
11.  $\text{Smith } k \{W\}_{\text{HR}}$  1, *a fortiori*
12.  $\text{Smith } k C$  7, *a fortiori*
13.  $\text{Smith } k C \wedge \text{Smith } k \{W\}_{\text{HR}}$  11, 12, PL
14.  $\text{Smith } k (C, \{W\}_{\text{HR}})$  13, pairing
15.  $\{W\}_{\text{HR}} :_{\text{Smith}}^{\emptyset} \text{job}(\text{Jones})$  6, *a fortiori*
16.  $(\{W\}_{\text{HR}} :_{\text{Smith}}^{\emptyset} \text{job}(\text{Jones})) \rightarrow (C, \{W\}_{\text{HR}}) :_{\text{Smith}}^{\emptyset} \text{job}(\text{Jones})$  p. ext.
17.  $(C, \{W\}_{\text{HR}}) :_{\text{Smith}}^{\emptyset} \text{job}(\text{Jones})$  15, 16, PL
18.  $(C, \{W\}_{\text{HR}}) :_{\text{Smith}}^{\emptyset} (\text{job}(\text{Jones}) \wedge 10(\text{Jones}))$  10, 17, proof conj.
19.  $\text{Smith } k (C, \{W\}_{\text{HR}}) \wedge$   
 $(C, \{W\}_{\text{HR}}) :_{\text{Smith}}^{\emptyset} (\text{job}(\text{Jones}) \wedge 10(\text{Jones}))$  14, 18, PL
20.  $C :_{\text{Smith}}^{\emptyset} 10(\text{Jones}) \rightarrow$   
 $\left( \begin{array}{l} \text{Smith } k (C, \{W\}_{\text{HR}}) \wedge \\ (C, \{W\}_{\text{HR}}) :_{\text{Smith}}^{\emptyset} (\text{job}(\text{Jones}) \wedge 10(\text{Jones})) \end{array} \right)$  8–19, LDT

$$\begin{array}{ll}
21. & \left( \begin{array}{l} \text{Smith } k C \rightarrow \\ \left( C :_{\text{Smith}}^{\emptyset} 10(\text{Jones}) \rightarrow \right. \\ \left. \left( \text{Smith } k (C, \{W\}_{\text{HR}}) \wedge \right. \right. \\ \left. \left. (C, \{W\}_{\text{HR}}) :_{\text{Smith}}^{\emptyset} (\text{job}(\text{Jones}) \wedge 10(\text{Jones})) \right) \right) \end{array} \right) \quad 7-20, \text{LDT} \\
22. & \left( \begin{array}{l} W :_{\text{HR}}^A \text{job}(\text{Jones}) \rightarrow \\ \text{Smith } k C \rightarrow \\ \left( C :_{\text{Smith}}^{\emptyset} 10(\text{Jones}) \rightarrow \right. \\ \left( \text{Smith } k (C, \{W\}_{\text{HR}}) \wedge \right. \\ \left. (C, \{W\}_{\text{HR}}) :_{\text{Smith}}^{\emptyset} (\text{job}(\text{Jones}) \wedge 10(\text{Jones})) \right) \end{array} \right) \quad 2-21, \text{LDT} \\
23. & \vdash_{\text{LiP}} \left( \begin{array}{l} \text{Smith } k \{W\}_{\text{HR}} \rightarrow \\ W :_{\text{HR}}^A \text{job}(\text{Jones}) \rightarrow \\ \text{Smith } k C \rightarrow \\ \left( C :_{\text{Smith}}^{\emptyset} 10(\text{Jones}) \rightarrow \right. \\ \left( \text{Smith } k (C, \{W\}_{\text{HR}}) \wedge \right. \\ \left. (C, \{W\}_{\text{HR}}) :_{\text{Smith}}^{\emptyset} (\text{job}(\text{Jones}) \wedge 10(\text{Jones})) \right) \end{array} \right) \quad 1-22, \\
& \text{LDT} \\
24. & \vdash_{\text{LiP}} \left( \begin{array}{l} (\text{Smith } k \{W\}_{\text{HR}} \wedge W :_{\text{HR}}^A \text{job}(\text{Jones})) \rightarrow \\ \wedge \text{Smith } k C \wedge C :_{\text{Smith}}^{\emptyset} 10(\text{Jones}) \end{array} \right) \rightarrow \quad 23, \text{PL.} \\
& \left( (C, \{W\}_{\text{HR}}) :_{\text{Smith}}^{\emptyset} (\text{job}(\text{Jones}) \wedge 10(\text{Jones})) \right)
\end{array}$$

□

In the preceding proof, observe the use of the law of proof extension, deducible by means of epistemic antitonicity, and expressing the monotonicity of LiP-proofs. Like Artëmov, who interprets Lehrer and Paxson's indefeasibility condition for justified true belief as possibly corresponding to LP's sum-axiom (cf. [Art08a]), we could thus interpret this condition as corresponding to LiP's proof extension.

**Corollary 5** (Gettier-example with signing). *Given  $\mathcal{A} := \{\text{Smith}, \text{Jones}, \text{HR}\}$ ,*

$$\begin{array}{l}
\{(\text{job}(\text{Smith}) \wedge \text{job}(\text{Jones})) \rightarrow \perp\}; \emptyset \vdash_{\text{LiP}} \\
\left( \text{Smith } k \{W\}_{\text{HR}} \wedge (W :_{\text{HR}}^A \text{job}(\text{Jones})) \right) \rightarrow \\
\left( \wedge \text{Smith } k C \wedge C :_{\text{Smith}}^{\emptyset} 10(\text{Jones}) \right) \rightarrow \\
((\text{job}(\text{Smith}) \wedge 10(\text{Smith})) \rightarrow \perp)
\end{array}$$

*Proof.* From Proposition 12 and Lemma 3. □

### 3.4 Multi-agent LP-like systems

By their quality of being conservative extensions of non-interactive LP-like systems, the following logical systems with multi-agent character inherit the lack of message-passing interactivity of LP in the following senses: namely the lack of (1) a sound truth axiom for message passing (cf. Section 3.1.1), (2) the transferability of local truths by means of messages (cf. Section 3.2.2), and (3) signature checking that could act as proof checking of claimed local truths (cf.



Section 3.1.2). In our understanding, these lacks of LP-like systems *without* message passing are reflected by the fact that LP can only homomorph but not embed into interactive-proof systems *with* message passing like LiP.

### 3.4.1 LP<sup>2</sup>

Yavorskaya’s LP<sup>2</sup> [Yav08] is an extension of LP with multi-agent character in the sense that LP<sup>2</sup> extends LP with a *2-agent* view such that each one of the two agents

1. has her own proof-sum, proof-application, and proof-checker constructor
2. may have a constructor for
  - (a) checking the other agent’s proofs, i.e., peer proofs
  - (b) converting peer proofs into proofs of her own.

LP<sup>2</sup> being an extension of LP, our criticism of LP also applies to LP<sup>2</sup>. Also, LiP can manage an  $n$ -agent view for arbitrary  $n \in \mathbb{N}$  with only  $n + 1$  (transmittable) proof-term constructors ( $n$  signature constructors plus 1 pair constructor). This feature is the fruit of our design decision to equip LiP with proof-term signature constructors and an agent-parametric proof modality, which allows the association as proof of arbitrary data to arbitrary verifying agents within arbitrary peer communities. Whereas an extension of LP<sup>2</sup> to LP <sup>$n$</sup>  for a fixed  $n \in \mathbb{N}$  would require  $3n + 2n(n - 1) = 2n^2 + n$  constructors ( $n$  proof-sum plus  $n$  proof-application plus  $n$  proof-checker plus  $n(n - 1)$  peer-proof-checker plus  $n(n - 1)$  peer-proof-conversion constructors), and still not allow the free association of proofs to agents. In sum, LiP seems more appropriate for interactivity and is even simpler than would be LP <sup>$n$</sup> . However, it could be interesting to parametrise Yavorskaya’s agent-centric proof converters with agent *communities* so that two communities that do not share their respective common knowledge of what should constitute a proof could communicate with each other thanks to such communal proof converters.

### 3.4.2 UL

Renne’s UL [Ren12] is an extension of Artëmov’s Justification Logic, JL [Art08a] (including Artëmov’s LP) with multi-agent character in the sense that UL combines JL with (multi-agent) Dynamic Epistemic Logic [vDvdHK07]. Of course, dynamic extensions of static logics are interesting. The sophisticated language of UL is defined by staged mutual recursion on the structure of terms and formulas, and has a semantic interface in the style of LP but crucially only over finite Kripke-models. The mutual recursion arises in the application term constructor of UL, which has a formula parameter meant to indicate the relevance of the second constructor argument to the constructor parameter in UL’s application axiom. Given that sum and application can be subsumed by pairing

in LiP (cf. Section 3.1.2), it would be interesting to experiment with a formula-parametrised pair constructor in UL intended to subsume sum and formula-parametrised application. The justification terms in UL do not provide evidence for knowledge but only for belief, which is expressed with a K4-modality. (Usually, belief is expressed with a KD45-modality [MV07].) The restriction of UL to interpretation in finite Kripke-models is not unproblematic. Communication systems (e.g., the Internet) can *not* in general be faithfully modelled as *finite* transition systems and thus *finite* Kripke-models.

## 4 Conclusion

### 4.1 Assessment

We have proposed a logic of interactive proofs with as main contributions those described in Section 1.4.1. Our resulting notion of proofs has the advantage of being not only operational thanks to a proof-theoretic definition but also *declarative* thanks to a complementary model-theoretic definition, which gives a *constructive epistemic semantics* to proofs in the sense of explicating *what* (knowledge) proofs effect in agents, complementing thereby the (operational) axiomatics, which explicates *how* proofs do so. In particular, first, *interactive* computation is *semantic* computation: we not only compute result values (syntax), but (knowledge) equivalence classes of them (semantics); and, second, our definition of interactive proofs reflects the impact of mathematical proofs in a social sense (cf. Section 1.2.1): if my peer knew my proof for her of a given statement then she would know that the statement is true. (Notice the different kinds of knowledge and the conditional mode!) In contrast, the traditional definition of (mathematical) proofs is only operational in the sense that proofs are defined purely in terms of the deductive operations that are used to construct them. Their *pragmatics*, i.e., their (epistemic) impact in proof-reading agents, was left unformalised, and their operational definition risks restricting their generality. However now thanks to our formalisation, we as a community have the formal common knowledge that

- agents in distributed systems are at the same time computation oracles, data miners, meaning interpreters, message-passing communicators, interactive provers, and logical combinators
- **a proof is *that which if known to one of our peer members would induce the knowledge of its proof goal with that member.***

### 4.2 Future work

Our future lines of research for LiP are the following:

1. present the other interactive structures and morphisms mentioned in Figure 1

2. study the computability of satisfiability and global and local model checking
3. develop the proof theory of LiP (alternative calculi, proof complexity), including a proof-theoretic explication of our instant interactive proofs
4. extend LiP with guarded quantifiers, dynamic modalities, and fixpoint operators (Hennessy-Milner correspondence, characteristic formulas)
5. extend LiP with the classical and the modern conception of cryptography mentioned in Footnote 8 (requiring resource-bounded agents)
6. apply LiP and its variants to the analysis and synthesis of communication protocols (proof-carrying code correct by construction via program extraction from constructive proofs of correctness, on-line interactive algorithms)
7. create the Logic of Evidence and the Logic of Deception suggested on Page 25.

Applying LiP means fixing four things if need be, namely, at the level of

1. *terms*:
  - (a) the choice of term axioms;
  - (b) the application-specific base data  $B$ ;
  - (c) the implementation of signing, e.g., in terms of public-key cryptography;
2. *formulas*: the set  $\mathcal{P}$  of atomic propositions (those besides  $a \text{ k } M$ ) together with the axioms governing their intended meaning.

This will instantiate LiP as a theory of the specific subject matter of the application, such as, for example, Dolev-Yao cryptography (cf. Page 12).

**Acknowledgements.** I thank Johan van Benthem and Larry Moss for their early encouragement and Sergei Artëmov for his sympathetic consideration of my work. I also thank Eiji Okamoto for being my generous host during my post-doctoral fellowship from the Japan Society for the Promotion of Science, Jean-Luc Beuchot for suggesting to apply for such a fellowship and for helping me with and sharing with me the fascinating life of a *gaijin*, and Shihoko Sekiya for being our perfect secretary and daily office sunshine. Last but not least, I thank Denis Saveliev for giving me his favourable opinion on this paper, and Olga Grinchtein for spotting and informing me about a few typos.

## References

- [AB05] S.N. Artëmov and L.D. Beklemishev. *Handbook of Philosophical Logic*, volume 12, chapter Provability Logic. Springer, second edition, 2005.
- [AN05] S. Artemov and E. Nogina. Introducing justification into epistemic logic. *Journal of Logic and Computation*, 15(6), 2005.
- [And08] R. Anderson. *Security Engineering: A Guide to Building Dependable Distributed Systems*. Wiley, second edition, 2008.
- [Art94] S. Artemov. Logic of proofs. *Annals of Pure and Applied Logic*, 67, 1994.
- [Art01] S. Artemov. Explicit provability and constructive semantics. *Bulletin of Symbolic Logic*, 7(1), 2001.
- [Art07] S. Artemov. *Handbook of Modal Logic*, chapter Modal Logic in Mathematics. Volume 3 of Blackburn et al. [BvBW07], 2007.
- [Art08a] S. Artemov. The logic of justifications. *The Review of Symbolic Logic*, 1(4), 2008.
- [Art08b] S. Artemov. Symmetric logic of proofs. In *Pillars of Computer Science: Essays Dedicated to Boris (Boaz) Trakhtenbrot on the Occasion of His 85th Birthday*, volume 4800 of LNCS. Springer, 2008.
- [BG07] T. Braüner and S. Ghilardi. *Handbook of Modal Logic*, chapter First-Order Modal Logic. Volume 3 of Blackburn et al. [BvBW07], 2007.
- [Bus98a] S.R. Buss. *Handbook of Proof Theory*, chapter An Introduction to Proof Theory. In [Bus98b], 1998.
- [Bus98b] S.R. Buss, editor. *Handbook of Proof Theory*. Elsevier, 1998.
- [BvB07] P. Blackburn and J. van Benthem. *Handbook of Modal Logic*, chapter Modal Logic: A Semantic Perspective. Volume 3 of Blackburn et al. [BvBW07], 2007.
- [BvBW07] P. Blackburn, J. van Benthem, and F. Wolter, editors. *Handbook of Modal Logic*, volume 3 of *Studies in Logic and Practical Reasoning*. Elsevier, 2007.
- [dG95] Ph. de Groote, editor. *The Curry-Howard Isomorphism*. Number 8 in Cahiers du centre de logique. Academia-Erasme, Louvain-la-Neuve (Belgique), 1995.

- [DP02] B.A. Davey and H.A. Priestley. *Introduction to Lattices and Order*. Cambridge University Press, 1990 (2002).
- [DY83] D. Dolev and A. Yao. On the security of public key protocols. *IEEE Transactions on Information Theory*, 29(12), 1983.
- [Fef89] S. Feferman. *The Number Systems: Foundations of Algebra and Analysis*. AMS Chelsea Publishing, second edition, 1964 (1989). Reprinted by the American Mathematical Society, 2003.
- [FHMV95] R. Fagin, J.Y. Halpern, Y. Moses, and M.Y. Vardi. *Reasoning about Knowledge*. MIT Press, 1995.
- [Fit05] M. Fitting. The logic of proofs, semantically. *Annals of Pure and Applied Logic*, 132, 2005.
- [Fit07] M. Fitting. *Handbook of Modal Logic*, chapter Modal Proof Theory. Volume 3 of Blackburn et al. [BvBW07], 2007.
- [Gab95] D.M. Gabbay, editor. *What Is a Logical System?* Number 4 in Studies in Logic and Computation. Oxford University Press, 1995.
- [Gol01] O. Goldreich. *Foundations of Cryptography: Basic Tools*. Cambridge University Press, 2001.
- [Gol04] O. Goldreich. *Foundations of Cryptography: Basic Applications*. Cambridge University Press, 2004.
- [GSW06] D. Goldin, S.A. Smolka, and P. Wegner, editors. *Interactive Computation: The New Paradigm*. Springer, 2006.
- [HM90] J.Y. Halpern and Y. Moses. Knowledge and common knowledge in a distributed environment. *Journal of the ACM*, 37(3), 1990.
- [HM92] J.Y. Halpern and Y. Moses. A guide to completeness and complexity for modal logics of knowledge and belief. *Artificial Intelligence*, 54(3), 1992.
- [HR10] V.F. Hendricks and O. Roy, editors. *Epistemic Logic: 5 Questions*. Automatic Press, 2010.
- [HS08] J.R. Hindley and J.P. Seldin. *Lambda-Calculus and Combinators*. Cambridge University Press, second edition, 2008.
- [JdJ98] G. Japaridze and D. de Jongh. *Handbook of Proof Theory*, chapter The Logic of Provability. In Buss [Bus98b], 1998.
- [Kat10] J. Katz. *Digital Signatures*. Springer, 2010.
- [KR10] S. Kramer and A. Rybalchenko. A multi-modal framework for achieving accountability in multi-agent systems. In *Proceedings of the ESSLLI-affiliated Workshop on Logics in Security*, 2010.

- [KR11] S. Kramer and P.Y.A. Ryan. A modular multi-modal specification of real-timed, end-to-end voter-verifiable voting systems. In *Proceedings of the RE-affiliated Workshop on Requirements Engineering for Electronic Voting Systems*. IEEE, 2011.
- [Kra07a] S. Kramer. *Logical Concepts in Cryptography*. PhD thesis, Ecole Polytechnique Fédérale de Lausanne, 2007. <http://library.epfl.ch/en/theses/?nr=3845>.
- [Kra07b] S. Kramer. The meaning of a cryptographic message via hypothetical knowledge and provability. In *Proceedings of the Workshop on Logic, Rationality and Interaction*, London, 2007. King’s College Publications.
- [Kra08a] S. Kramer. Cryptographic Protocol Logic: Satisfaction for (timed) Dolev-Yao cryptography. *Journal of Logic and Algebraic Programming*, 77(1–2), 2008. <http://dx.doi.org/10.1016/j.jlap.2008.05.005>.
- [Kra08b] S. Kramer. Reducing provability to knowledge in multi-agent systems. In *Proceedings of the LiCS-affiliated Intuitionistic Modal Logics and Applications Workshop*, 2008. <ftp://ftp.research.microsoft.com/pub/tr/TR-2008-90.pdf>.
- [Kra10a] S. Kramer. An intuitionistic foundation for interactive computation. In *Abstract and Handout Booklet of the Conference on Computability in Europe: Programs, Proofs, Processes*. University of Azores, 2010. short paper.
- [Kra10b] S. Kramer. An intuitionistic foundation for interactive computation. In *Proceedings of the Conference on Advances in Modal Logic*, 2010. short paper.
- [Kuz00] R. Kuznets. On the complexity of explicit modal logics. volume 1862 of *LNCS*. Springer, 2000.
- [Mos06] Y. Moschovakis. *Notes on Set Theory*. Springer, 2nd edition, 2006.
- [MV07] J.-J. Meyer and F. Veltman. *Handbook of Modal Logic*, chapter Intelligent Agents and Common Sense Reasoning. Volume 3 of Blackburn et al. [BvBW07], 2007.
- [PR03] R. Parikh and R. Ramanujam. A knowledge based semantics of messages. *Journal of Logic, Language and Information*, 12, 2003.
- [Ren12] B. Renne. Multi-agent Justification Logic: communication and evidence elimination. *Synthese*, 185, Supplement 1, 2012.
- [SD08] D.A. Simovici and C. Djeraba. *Mathematical Tools for Data Mining: Set Theory, Partial Orders, Combinatorics*. Springer, 2008.

- [Sha48] C.E. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27, 1948.
- [Sha49] C.E. Shannon. Communication theory of secrecy systems. *Bell System Technical Journal*, 28, 1949.
- [Tay99] P. Taylor. *Practical Foundations of Mathematics*. Cambridge University Press, 1999.
- [TGD10] A. Tiu, R. Goré, and J. Dawson. A proof theoretic analysis of intruder theories. *Logical Methods in Computer Science*, 6(3), 2010.
- [vB97] J. van Benthem. *Logic and Reality: Essays on the Legacy of Arthur Prior*, chapter Modal Logic as a Theory of Information. Clarendon Press, Oxford, 1997.
- [vB09] J. van Benthem. The information in intuitionistic logic. *Synthese*, 167, 2009.
- [vDvdHK07] H. van Ditmarsch, W. van der Hoek, and B. Kooi. *Dynamic Epistemic Logic*, volume 337 of *Synthese Library*. Springer, 2007.
- [Ven07] Y. Venema. *Handbook of Modal Logic*, chapter Algebras and Coalgebras. Volume 3 of Blackburn et al. [BvBW07], 2007.
- [Yav08] T. Yavorskaya (Sidon). Interacting explicit evidence systems. *Theory of Computing Systems*, 43(2), 2008.

## A Completeness proof

**Completeness** For all  $\phi \in \mathcal{L}$ , if  $\models \phi$  then  $\vdash_{\text{LiP}} \phi$ .

*Proof.* Let

- $\mathcal{W}$  designate the set of all maximally LiP-consistent sets<sup>14</sup>
- for all  $w, w' \in \mathcal{W}$ ,  $w \mathrel{MC_a^C} w' : \text{iff } \{ \phi \in \mathcal{L} \mid M :_a^C \phi \in w \} \subseteq w'$
- for all  $w \in \mathcal{W}$ ,  $w \in \mathcal{V}_C(P) : \text{iff } P \in w$ .

---

<sup>14\*</sup> A set  $W$  of LiP-formulas is maximally LiP-consistent :iff  $W$  is LiP-consistent and  $W$  has no proper superset that is LiP-consistent. A set  $W$  of LiP-formulas is LiP-consistent :iff  $W$  is not LiP-inconsistent. A set  $W$  of LiP-formulas is LiP-inconsistent :iff there is a finite  $W' \subseteq W$  such that  $((\bigwedge W') \rightarrow \perp) \in \text{LiP}$ . Any LiP-consistent set can be extended to a maximally LiP-consistent set by means of the Lindenbaum Construction [Fit07, Page 90]. A set is maximally LiP-consistent if and only if the set of logical-equivalence classes of the set is an ultrafilter of the Lindenbaum-Tarski algebra of LiP [Ven07, Page 351]. The canonical frame is isomorphic to the ultrafilter frame of that Lindenbaum-Tarski algebra [Ven07, Page 352].

Then  $\mathfrak{M}_C := (\mathcal{W}, \{MC_a^C\}_{M \in \mathcal{M}, a \in \mathcal{A}, C \subseteq \mathcal{A}}, \mathcal{V}_C)$  designates the *canonical model* for LiP. Following Fitting [Fit07, Section 2.2], the following useful property of  $\mathfrak{M}_C$ ,

$$\boxed{\text{for all } \phi \in \mathcal{L} \text{ and } w \in \mathcal{W}, \phi \in w \text{ if and only if } \mathfrak{M}_C, w \models \phi,}$$

the so-called *Truth Lemma*, can be proved by induction on the structure of  $\phi$ :

1. Base case ( $\phi := P$  for  $P \in \mathcal{P}$ ). For all  $w \in \mathcal{W}$ ,  $P \in w$  if and only if  $\mathfrak{M}_C, w \models P$ , by definition of  $\mathcal{V}_C$ .
2. Inductive step ( $\phi := \neg\phi'$  for  $\phi' \in \mathcal{L}$ ). Suppose that for all  $w \in \mathcal{W}$ ,  $\phi' \in w$  if and only if  $\mathfrak{M}_C, w \models \phi'$ . Further let  $w \in \mathcal{W}$ . Then,  $\neg\phi' \in w$  if and only if  $\phi' \notin w$  —  $w$  is consistent — if and only if  $\mathfrak{M}_C, w \not\models \phi'$  — by the induction hypothesis — if and only if  $\mathfrak{M}_C, w \models \neg\phi'$ .
3. Inductive step ( $\phi := \phi' \wedge \phi''$  for  $\phi', \phi'' \in \mathcal{L}$ ). Suppose that for all  $w \in \mathcal{W}$ ,  $\phi' \in w$  if and only if  $\mathfrak{M}_C, w \models \phi'$ , and that for all  $w \in \mathcal{W}$ ,  $\phi'' \in w$  if and only if  $\mathfrak{M}_C, w \models \phi''$ . Further let  $w \in \mathcal{W}$ . Then,  $\phi' \wedge \phi'' \in w$  if and only if  $(\phi' \in w \text{ and } \phi'' \in w)$ , because  $w$  is maximal. Now suppose that  $\phi' \in w$  and  $\phi'' \in w$ . Hence,  $\mathfrak{M}_C, w \models \phi'$  and  $\mathfrak{M}_C, w \models \phi''$ , by the induction hypotheses, and thus  $\mathfrak{M}_C, w \models \phi' \wedge \phi''$ . Conversely, suppose that  $\mathfrak{M}_C, w \models \phi' \wedge \phi''$ . Then,  $\mathfrak{M}_C, w \models \phi'$  and  $\mathfrak{M}_C, w \models \phi''$ . Hence,  $\phi' \in w$  and  $\phi'' \in w$ , by the induction hypotheses. Thus,  $(\phi' \in w \text{ and } \phi'' \in w)$  if and only if  $(\mathfrak{M}_C, w \models \phi' \text{ and } \mathfrak{M}_C, w \models \phi'')$ . Whence  $\phi' \wedge \phi'' \in w$  if and only if  $(\mathfrak{M}_C, w \models \phi' \text{ and } \mathfrak{M}_C, w \models \phi'')$ , by transitivity.
4. Inductive step ( $\phi := M :_a^C \phi'$  for  $M \in \mathcal{M}$ ,  $a \in \mathcal{A}$ ,  $C \subseteq \mathcal{A}$ , and  $\phi' \in \mathcal{L}$ ).
  - 4.1 for all  $w \in \mathcal{W}$ ,  $\phi' \in w$  if and only if  $\mathfrak{M}_C, w \models \phi'$  ind. hyp.
  - 4.2  $w \in \mathcal{W}$  hyp.
  - 4.3  $M :_a^C \phi' \in w$  hyp.
  - 4.4  $w' \in \mathcal{W}$  hyp.
  - 4.5  $w MC_a^C w'$  hyp.
  - 4.6  $\{ \phi'' \in \mathcal{L} \mid M :_a^C \phi'' \in w \} \subseteq w'$  4.5
  - 4.7  $\phi' \in \{ \phi'' \in \mathcal{L} \mid M :_a^C \phi'' \in w \}$  4.3, 4.6
  - 4.8  $\phi' \in w'$  4.6, 4.7
  - 4.9  $\mathfrak{M}_C, w' \models \phi'$  4.1, 4.4, 4.8
  - 4.10 if  $w MC_a^C w'$  then  $\mathfrak{M}_C, w' \models \phi'$  4.5–4.9
  - 4.11 for all  $w' \in \mathcal{W}$ , if  $w MC_a^C w'$  then  $\mathfrak{M}_C, w' \models \phi'$  4.4–4.10
  - 4.12  $\mathfrak{M}_C, w \models M :_a^C \phi'$  4.11
  - 4.13  $M :_a^C \phi' \notin w$  hyp.
  - 4.14  $\mathcal{F} = \{ \phi'' \in \mathcal{L} \mid M :_a^C \phi'' \in w \} \cup \{ \neg\phi' \}$  hyp.
  - 4.15  $\mathcal{F}$  is LiP-inconsistent hyp.



4.16	there is $\{M :_a^C \phi_1, \dots, M :_a^C \phi_n\} \subseteq w$ such that	
	$\vdash_{\text{LiP}} (\phi_1 \wedge \dots \wedge \phi_n \wedge \neg \phi') \rightarrow \perp$	4.14, 4.15
4.17	$\{M :_a^C \phi_1, \dots, M :_a^C \phi_n\} \subseteq w$ and	
	$\vdash_{\text{LiP}} (\phi_1 \wedge \dots \wedge \phi_n \wedge \neg \phi') \rightarrow \perp$	hyp.
4.18	$\vdash_{\text{LiP}} (\phi_1 \wedge \dots \wedge \phi_n) \rightarrow \phi'$	4.17
4.19	$\vdash_{\text{LiP}} (M :_a^C (\phi_1 \wedge \dots \wedge \phi_n)) \rightarrow M :_a^C \phi'$	4.18, regularity
4.20	$\vdash_{\text{LiP}} ((M :_a^C \phi_1) \wedge \dots \wedge (M :_a^C \phi_n)) \rightarrow M :_a^C \phi'$	4.19
4.21	$M :_a^C \phi' \in w$	4.17, 4.20, $w$ is maximal
4.22	false	4.13, 4.21
4.23	false	4.16, 4.17–4.22
4.24	$\mathcal{F}$ is LiP-consistent	4.15–4.23
4.25	there is $w' \supseteq \mathcal{F}$ s.t. $w'$ is maximally LiP-consistent	4.24
4.26	$\mathcal{F} \subseteq w'$ and $w'$ is maximally LiP-consistent	hyp.
4.27	$\{ \phi'' \in \mathcal{L} \mid M :_a^C \phi'' \in w \} \subseteq \mathcal{F}$	4.14
4.28	$\{ \phi'' \in \mathcal{L} \mid M :_a^C \phi'' \in w \} \subseteq w'$	4.26, 4.27
4.29	$w \text{ } {}_M C_a^C w'$	4.28
4.30	$w' \in \mathcal{W}$	4.26
4.31	$\neg \phi' \in \mathcal{F}$	4.14
4.32	$\neg \phi' \in w'$	4.26, 4.31
4.33	$\phi' \notin w'$	4.26 ( $w'$ is LiP-consistent), 4.32
4.34	$\mathfrak{M}_{\mathcal{C}}, w' \not\models \phi'$	4.1, 4.33
4.35	there is $w' \in \mathcal{W}$ s.t. $w \text{ } {}_M C_a^C w'$ and $\mathfrak{M}_{\mathcal{C}}, w' \not\models \phi'$	4.29, 4.34
4.36	$\mathfrak{M}_{\mathcal{C}}, w \not\models M :_a^C \phi'$	4.35
4.37	$\mathfrak{M}_{\mathcal{C}}, w \not\models M :_a^C \phi'$	4.25, 4.26–4.36
4.38	$\mathfrak{M}_{\mathcal{C}}, w \not\models M :_a^C \phi'$	4.14–4.37
4.39	$M :_a^C \phi' \in w$ if and only if $\mathfrak{M}_{\mathcal{C}}, w \models M :_a^C \phi'$	4.3–4.12, 4.13–4.38
4.40	for all $w \in \mathcal{W}$ , $M :_a^C \phi' \in w$ if and only if $\mathfrak{M}_{\mathcal{C}}, w \models M :_a^C \phi'$	4.2–4.39

With the Truth Lemma we can now prove that for all  $\phi \in \mathcal{L}$ , if  $\not\vdash_{\text{LiP}} \phi$  then  $\not\models \phi$ . Let  $\phi \in \mathcal{L}$ , and suppose that  $\not\vdash_{\text{LiP}} \phi$ . Thus,  $\{\neg \phi\}$  is LiP-consistent, and can be extended to a maximally LiP-consistent set  $w$ , i.e.,  $\neg \phi \in w \in \mathcal{W}$ . Hence  $\mathfrak{M}_{\mathcal{C}}, w \models \neg \phi$ , by the Truth Lemma. Thus:  $\mathfrak{M}_{\mathcal{C}}, w \not\models \phi$ ,  $\mathfrak{M}_{\mathcal{C}} \not\models \phi$ , and  $\not\models \phi$ . That is,  $\mathfrak{M}_{\mathcal{C}}$  is a *universal* (for all  $\phi \in \mathcal{L}$ ) *counter-model* (if  $\phi$  is a non-theorem then  $\mathfrak{M}_{\mathcal{C}}$  falsifies  $\phi$ ).

We are left to prove that  $\mathfrak{M}_{\mathcal{C}}$  is also an *LiP-model*. So let us instantiate our data mining operator  $\text{cl}_a$  (cf. Page 18) on  $\mathcal{W}$  by letting for all  $w \in \mathcal{W}$

$$\text{msgs}_a(w) := \{ M \mid a \text{ k } M \in w \},$$

and let us prove that:

1. (a) for all  $w \in \mathcal{W}$ , if  $M \in \text{cl}_a^w(\emptyset)$  then  $w \text{ } {}_M\text{C}_a^{\mathcal{C}} w$   
(b) for all  $\mathcal{C}' \subseteq \mathcal{A}$ , if  $\mathcal{C} \subseteq \mathcal{C}'$  then  ${}_M\text{C}_a^{\mathcal{C}} \subseteq {}_M\text{C}_a^{\mathcal{C}'}$   
(c) for all  $w, w', w'' \in \mathcal{W}$ , if  $w \text{ } {}_M\text{C}_a^{\mathcal{C}} w'$  and  $w' \text{ } {}_M\text{C}_a^{\mathcal{C}} w''$  then  $w \text{ } {}_M\text{C}_a^{\mathcal{C}} w''$
2. for all  $w \in \mathcal{W}$ :  
(a) if (for all  $w' \in \mathcal{W}$ ,  $M \in \text{cl}_a^{w'}(\emptyset)$  implies  $M' \in \text{cl}_a^{w'}(\emptyset)$ ) then  ${}_w\llbracket M' \rrbracket_a^{\mathcal{C}} \subseteq {}_w\llbracket M \rrbracket_a^{\mathcal{C}}$   
(b) if  $\phi \rightarrow \phi' \in {}_w\llbracket M \rrbracket_a^{\mathcal{C}}$  and  $\phi \in {}_w\llbracket M' \rrbracket_a^{\mathcal{C}}$  then  $\phi' \in {}_w\llbracket (M, M') \rrbracket_a^{\mathcal{C}}$   
(c) if  $\phi \in {}_w\llbracket M \rrbracket_a^{\mathcal{C}}$  then for all  $b \in \mathcal{C} \cup \{a\}$ ,  $a \text{ k } M \wedge M :_a^{\mathcal{C}} \phi \in {}_w\llbracket \{M\} \rrbracket_b^{\mathcal{C} \cup \{a\}}$

where

$${}_w\llbracket M \rrbracket_a^{\mathcal{C}} := \{ \phi \in \mathcal{L} \mid \mathfrak{M}_{\mathcal{C}}, w \models M :_a^{\mathcal{C}} \phi \}.$$

For (1.a), let  $w \in \mathcal{W}$  and suppose that  $M \in \text{cl}_a^w(\emptyset)$ . Hence  $a \text{ k } M \in w$  due to the maximality of  $w$ , which contains all the term axioms corresponding to the defining clauses of  $\text{cl}_a^w$ . Further suppose that  $M :_a^{\mathcal{C}} \phi \in w$ . Since  $w$  is maximal,

$$(M :_a^{\mathcal{C}} \phi) \rightarrow (a \text{ k } M \rightarrow \phi) \in w \quad (\text{epistemic truthfulness}).$$

Hence,  $a \text{ k } M \rightarrow \phi \in w$ , and  $\phi \in w$ , by consecutive *modus ponens*.

For (1.b), let  $\mathcal{C}' \subseteq \mathcal{A}$  and suppose that  $\mathcal{C} \subseteq \mathcal{C}'$ . That is,  $\mathcal{C} \cup \mathcal{C}' = \mathcal{C}'$ . Further, let  $w, w' \in \mathcal{W}$  and suppose that  $w \text{ } {}_M\text{C}_a^{\mathcal{C}} w'$ . That is, for all  $\phi \in \mathcal{L}$ , if  $(M :_a^{\mathcal{C}} \phi) \in w$  then  $\phi \in w'$ . Furthermore, let  $\phi \in \mathcal{L}$  and suppose that  $(M :_a^{\mathcal{C}'} \phi) \in w$ . Thus  $(M :_a^{\mathcal{C} \cup \mathcal{C}'} \phi) \in w$ . Since  $w$  is maximal,

$$(M :_a^{\mathcal{C} \cup \mathcal{C}'} \phi) \rightarrow M :_a^{\mathcal{C}} \phi \in w \quad (\text{group decomposition}).$$

Hence  $(M :_a^{\mathcal{C}} \phi) \in w$  by *modus ponens*, and thus  $\phi \in w'$ .

For (1.c), let  $w, w', w'' \in \mathcal{S}$  and suppose that  $w \text{ } {}_M\text{C}_a^{\mathcal{C}} w'$  and  $w' \text{ } {}_M\text{C}_a^{\mathcal{C}} w''$ . Further suppose that  $M :_a^{\mathcal{C}} \phi \in w$ . Since  $w$  is maximal,

$$(M :_a^{\mathcal{C}} \phi) \rightarrow \bigwedge_{b \in \mathcal{C} \cup \{a\}} (\llbracket M \rrbracket_a :_b^{\mathcal{C} \cup \{a\}} (M :_a^{\mathcal{C}} \phi)) \in w \quad (\text{peer review, short}).$$

Hence  $\bigwedge_{b \in \mathcal{C} \cup \{a\}} (\llbracket M \rrbracket_a :_b^{\mathcal{C} \cup \{a\}} (M :_a^{\mathcal{C}} \phi)) \in w$  by *modus ponens*, and thus for all  $b \in \mathcal{C} \cup \{a\}$ ,  $\llbracket M \rrbracket_a :_b^{\mathcal{C} \cup \{a\}} (M :_a^{\mathcal{C}} \phi) \in w$ , in particular  $\llbracket M \rrbracket_a :_a^{\mathcal{C} \cup \{a\}} (M :_a^{\mathcal{C}} \phi) \in w$ . Since  $w$  is maximal,

$$(\llbracket M \rrbracket_a :_a^{\mathcal{C} \cup \{a\}} (M :_a^{\mathcal{C}} \phi)) \rightarrow \llbracket M \rrbracket_a :_a^{\mathcal{C}} (M :_a^{\mathcal{C}} \phi) \in w \quad (\text{group decomposition}).$$

Hence  $\llbracket M \rrbracket_a :_a^{\mathcal{C}} (M :_a^{\mathcal{C}} \phi) \in w$  by *modus ponens*. Since  $w$  is maximal,

$$(\llbracket M \rrbracket_a :_a^{\mathcal{C}} (M :_a^{\mathcal{C}} \phi)) \rightarrow M :_a^{\mathcal{C}} (M :_a^{\mathcal{C}} \phi) \in w \quad (\text{self-signing elimination}).$$

Hence  $M :_a^C (M :_a^C \phi) \in w$  by *modus ponens*. Hence,  $M :_a^C \phi \in w'$  by  $w MC_a^C w'$ , and then  $\phi \in w''$  by  $w' MC_a^C w''$ .

For the rest, let  $w \in \mathcal{W}$ .

For (2.a), suppose that for all  $w' \in \mathcal{W}$ ,  $M \in cl_a^{w'}(\emptyset)$  implies  $M' \in cl_a^{w'}(\emptyset)$ . Hence for all  $w' \in \mathcal{W}$ ,  $a \mathbf{k} M \in w'$  implies  $a \mathbf{k} M' \in w'$  due to the maximality of  $w'$ , which contains all the term axioms corresponding to the defining clauses of  $cl_a^{w'}$ . Hence for all  $w' \in \mathcal{W}$ ,  $\mathfrak{M}_C, w' \models a \mathbf{k} M$  implies  $\mathfrak{M}_C, w' \models a \mathbf{k} M'$ , by the Truth Lemma. Thus for all  $w' \in \mathcal{W}$ ,  $\mathfrak{M}_C, w' \models a \mathbf{k} M \rightarrow a \mathbf{k} M'$ . Hence for all  $w' \in \mathcal{W}$ ,  $a \mathbf{k} M \rightarrow a \mathbf{k} M' \in w'$  by the Truth Lemma. Hence also:

- for all  $w' \in \mathcal{W}$ ,  $(M' :_a^C \phi) \rightarrow M :_a^C \phi \in w'$  by the universality of the canonical model and epistemic antitonicity
- in particular,  $(M' :_a^C \phi) \rightarrow M :_a^C \phi \in w$ .

Further, let  $\phi \in \mathcal{L}$  and suppose that  $\phi \in {}_w[M']_a^C$ . Thus,  $\mathfrak{M}_C, w \models M' :_a^C \phi$  by definition, and  $M' :_a^C \phi \in w$  by the Truth Lemma. Hence  $M :_a^C \phi \in w$  by *modus ponens*. Thus  $\mathfrak{M}_C, w \models M :_a^C \phi$  by the Truth Lemma, and thus  $\phi \in {}_w[M]_a^C$  by definition.

For (2.b), suppose that  $\phi \rightarrow \phi' \in {}_w[M]_a^C$  and  $\phi \in {}_w[M']_a^C$ . Thus,  $\mathfrak{M}_C, w \models M :_a^C (\phi \rightarrow \phi')$  and  $\mathfrak{M}_C, w \models M' :_a^C \phi$ , by definition. Hence,  $M :_a^C (\phi \rightarrow \phi') \in w$  and  $M' :_a^C \phi \in w$ , by the Truth Lemma. Since  $w$  is maximal,

$$(M :_a^C (\phi \rightarrow \phi')) \rightarrow ((M' :_a^C \phi) \rightarrow (M, M') :_a^C \phi') \in w \quad (\text{generalised Kripke-law}).$$

Hence,  $(M' :_a^C \phi) \rightarrow (M, M') :_a^C \phi' \in w$ , and  $(M, M') :_a^C \phi' \in w$ , by consecutive *modus ponens*. Thus  $\mathfrak{M}_C, w \models (M, M') :_a^C \phi'$  by the Truth Lemma, and  $\phi' \in {}_w[(M, M')]_a^C$  by definition.

For (2.c) suppose that  $\phi \in {}_w[M]_a^C$  and  $b \in \mathcal{C} \cup \{a\}$ . Hence,  $\mathfrak{M}_C, w \models M :_a^C \phi$  by definition, and  $M :_a^C \phi \in w$  by the Truth Lemma. Let  $w' \in \mathcal{W}$  and suppose that  $w \llbracket M \rrbracket_a^{C \cup \{a\}} w'$ . Thus  $\{ \phi \in \mathcal{L} \mid \llbracket M \rrbracket_a^{C \cup \{a\}} \phi \in w \} \subseteq w'$  by definition. Since  $w$  is maximal,

$$(M :_a^C \phi) \rightarrow \bigwedge_{b \in \mathcal{C} \cup \{a\}} (\llbracket M \rrbracket_a^{C \cup \{a\}} (a \mathbf{k} M \wedge M :_a^C \phi)) \in w \quad (\text{peer review}).$$

Hence,  $\bigwedge_{b \in \mathcal{C} \cup \{a\}} (\llbracket M \rrbracket_a^{C \cup \{a\}} (a \mathbf{k} M \wedge M :_a^C \phi)) \in w$ , and  $\llbracket M \rrbracket_a^{C \cup \{a\}} (a \mathbf{k} M \wedge M :_a^C \phi) \in w$ , by consecutive *modus ponens*; and hence  $a \mathbf{k} M \wedge M :_a^C \phi \in w'$ . Thus  $\mathfrak{M}_C, w' \models a \mathbf{k} M \wedge M :_a^C \phi$  by the Truth Lemma. Discharging hypotheses,  $\mathfrak{M}_C, w \models \llbracket M \rrbracket_a^{C \cup \{a\}} (a \mathbf{k} M \wedge M :_a^C \phi)$ . Thus  $a \mathbf{k} M \wedge M :_a^C \phi \in {}_w[\llbracket M \rrbracket_a^{C \cup \{a\}}]$  by definition.  $\square$

## B Other proofs

### B.1 Proof of Theorem 1

Let “PT” abbreviate “propositional tautology” and “PL” “propositional logic”, and let PT and PL refer to the propositional fragment of LiP only.

1. (a)  $\vdash_{\text{LiP}} a \mathbf{k} (M, M') \rightarrow (a \mathbf{k} M \wedge a \mathbf{k} M')$  unpairing  
(b)  $\vdash_{\text{LiP}} (a \mathbf{k} M \wedge a \mathbf{k} M') \rightarrow a \mathbf{k} M$  PT  
(c)  $\vdash_{\text{LiP}} a \mathbf{k} (M, M') \rightarrow a \mathbf{k} M$  a, b, PL.
2. Symmetrically to 1.
3. (a)  $\vdash_{\text{LiP}} a \mathbf{k} (M, M) \rightarrow a \mathbf{k} M$  left or right projection  
(b)  $\vdash_{\text{LiP}} a \mathbf{k} M \rightarrow (a \mathbf{k} M \wedge a \mathbf{k} M)$  PT  
(c)  $\vdash_{\text{LiP}} (a \mathbf{k} M \wedge a \mathbf{k} M) \rightarrow a \mathbf{k} (M, M)$  pairing  
(d)  $\vdash_{\text{LiP}} a \mathbf{k} M \rightarrow a \mathbf{k} (M, M)$  b, c, PL  
(e)  $\vdash_{\text{LiP}} a \mathbf{k} (M, M) \leftrightarrow a \mathbf{k} M$  a, d, PL.
4. (a)  $\vdash_{\text{LiP}} (a \mathbf{k} M \wedge a \mathbf{k} M') \leftrightarrow a \mathbf{k} (M, M')$  [un]pairing  
(b)  $\vdash_{\text{LiP}} (a \mathbf{k} M' \wedge a \mathbf{k} M) \leftrightarrow a \mathbf{k} (M', M)$  [un]pairing  
(c)  $\vdash_{\text{LiP}} (a \mathbf{k} M \wedge a \mathbf{k} M') \leftrightarrow (a \mathbf{k} M' \wedge a \mathbf{k} M)$  PT  
(d)  $\vdash_{\text{LiP}} a \mathbf{k} (M, M') \leftrightarrow a \mathbf{k} (M', M)$  a, b, c, PL.
5. (a)  $\vdash_{\text{LiP}} (a \mathbf{k} M \rightarrow a \mathbf{k} M') \leftrightarrow (a \mathbf{k} M \rightarrow (a \mathbf{k} M \wedge a \mathbf{k} M'))$  PT  
(b)  $\vdash_{\text{LiP}} (a \mathbf{k} M \wedge a \mathbf{k} M') \leftrightarrow a \mathbf{k} (M, M')$  [un]pairing  
(c)  $\vdash_{\text{LiP}} (a \mathbf{k} M \rightarrow a \mathbf{k} M') \leftrightarrow (a \mathbf{k} M \rightarrow a \mathbf{k} (M, M'))$  a, b, PL  
(d)  $\vdash_{\text{LiP}} a \mathbf{k} (M, M') \rightarrow a \mathbf{k} M$  left projection  
(e)  $\vdash_{\text{LiP}} (a \mathbf{k} M \rightarrow a \mathbf{k} M') \leftrightarrow (a \mathbf{k} (M, M') \leftrightarrow a \mathbf{k} M)$  c, d, PL.
6. (a)  $\vdash_{\text{LiP}} a \mathbf{k} a$  knowledge of one's own name  
(b)  $\vdash_{\text{LiP}} a \mathbf{k} a \rightarrow (a \mathbf{k} M \rightarrow a \mathbf{k} a)$  PT  
(c)  $\vdash_{\text{LiP}} a \mathbf{k} M \rightarrow a \mathbf{k} a$  b, c, PL  
(d)  $\vdash_{\text{LiP}} (a \mathbf{k} M \rightarrow a \mathbf{k} a) \leftrightarrow (a \mathbf{k} (M, a) \leftrightarrow a \mathbf{k} M)$  neutral pair elements  
(e)  $\vdash_{\text{LiP}} a \mathbf{k} (M, a) \leftrightarrow a \mathbf{k} M$  c, d, PL.
7. (a)  $\vdash_{\text{LiP}} (a \mathbf{k} M \wedge a \mathbf{k} (M', M'')) \leftrightarrow a \mathbf{k} (M, (M', M''))$  [un]pairing  
(b)  $\vdash_{\text{LiP}} (a \mathbf{k} M' \wedge a \mathbf{k} M'') \leftrightarrow a \mathbf{k} (M', M'')$  [un]pairing  
(c)  $\vdash_{\text{LiP}} (a \mathbf{k} M \wedge (a \mathbf{k} M' \wedge a \mathbf{k} M'')) \leftrightarrow a \mathbf{k} (M, (M', M''))$  a, b, PL  
(d)  $\vdash_{\text{LiP}} (a \mathbf{k} M \wedge (a \mathbf{k} M' \wedge a \mathbf{k} M'')) \leftrightarrow ((a \mathbf{k} M \wedge a \mathbf{k} M') \wedge a \mathbf{k} M'')$  PT  
(e)  $\vdash_{\text{LiP}} ((a \mathbf{k} M \wedge a \mathbf{k} M') \wedge a \mathbf{k} M'') \leftrightarrow a \mathbf{k} (M, (M', M''))$  c, d, PL  
(f)  $\vdash_{\text{LiP}} (a \mathbf{k} M \wedge a \mathbf{k} M') \leftrightarrow a \mathbf{k} (M, M')$  [un]pairing  
(g)  $\vdash_{\text{LiP}} (a \mathbf{k} (M, M') \wedge a \mathbf{k} M'') \leftrightarrow a \mathbf{k} (M, (M', M''))$  e, f, PL  
(h)  $\vdash_{\text{LiP}} (a \mathbf{k} (M, M') \wedge a \mathbf{k} M'') \leftrightarrow a \mathbf{k} ((M, M'), M'')$  [un]pairing  
(i)  $\vdash_{\text{LiP}} a \mathbf{k} (M, (M', M'')) \leftrightarrow a \mathbf{k} ((M, M'), M'')$  g, h, PL.
8. By propositional logic and epistemic antitonicity.

- 9–10 and 17 follow directly from epistemic antitonicity and the corresponding pairing laws by propositional logic.
11. By propositional logic directly from proof extension left and right.
- 12–13 and 15–16 follow directly from epistemic bitonicity and the corresponding pairing laws by propositional logic.
14. (a)  $\vdash_{\text{LiP}} a \mathbf{k} M \rightarrow a \mathbf{k} M'$  hypothesis  
 (b)  $\vdash_{\text{LiP}} (a \mathbf{k} M \rightarrow a \mathbf{k} M') \leftrightarrow (a \mathbf{k} (M, M') \leftrightarrow a \mathbf{k} M)$  neutral pair elements  
 (c)  $\vdash_{\text{LiP}} a \mathbf{k} (M, M') \leftrightarrow a \mathbf{k} M$  a, b, PL  
 (d)  $\{a \mathbf{k} (M, M') \leftrightarrow a \mathbf{k} M\} \vdash_{\text{LiP}} (M, M') :_a^C \phi \leftrightarrow M :_a^C \phi$  epistemic bitonicity  
 (e)  $\vdash_{\text{LiP}} (M, M') :_a^C \phi \leftrightarrow M :_a^C \phi$  c, d, PL  
 (f)  $\{a \mathbf{k} M \rightarrow a \mathbf{k} M'\} \vdash_{\text{LiP}} (M, M') :_a^C \phi \leftrightarrow M :_a^C \phi$  a–e, definition.
18. (a)  $\vdash_{\text{LiP}} ((M :_a^C \phi) \vee b :_a^C \phi) \rightarrow (M, b) :_a^C \phi$  proof extension  
 (b)  $\vdash_{\text{LiP}} a \mathbf{k} \llbracket M \rrbracket_b \rightarrow a \mathbf{k} (M, b)$  signature analysis  
 (c)  $\vdash_{\text{LiP}} ((M, b) :_a^C \phi) \rightarrow \llbracket M \rrbracket_b :_a^C \phi$  b, epistemic antitonicity  
 (d)  $\vdash_{\text{LiP}} ((M :_a^C \phi) \vee b :_a^C \phi) \rightarrow \llbracket M \rrbracket_b :_a^C \phi$  a, c, PL.
19. (a)  $\vdash_{\text{LiP}} (\llbracket M \rrbracket_a :_a^C \phi) \rightarrow M :_a^C \phi$  self-signing elimination  
 (b)  $\vdash_{\text{LiP}} ((M :_a^C \phi) \vee b :_a^C \phi) \rightarrow \llbracket M \rrbracket_a :_a^C \phi$  signing introduction  
 (c)  $\vdash_{\text{LiP}} (M :_a^C \phi) \rightarrow \llbracket M \rrbracket_a :_a^C \phi$  b, PL  
 (d)  $\vdash_{\text{LiP}} (\llbracket M \rrbracket_a :_a^C \phi) \leftrightarrow M :_a^C \phi$  a, c, PL.
20. Suppose that  $\mathcal{A} = \{a\}$ .
- (a) Let us proceed by induction over  $M \in \mathcal{M}$ .
- i. *base case*, i.e.,  $M := b$ , for  $b \in \mathcal{A}$ . Hence  $b = a$ , and thus  $\vdash_{\text{LiP}} a \mathbf{k} b$  because  $\vdash_{\text{LiP}} a \mathbf{k} a$ .
  - ii. *inductive step* for  $M := \llbracket M' \rrbracket_b$ , for  $M' \in \mathcal{M}$  and  $b \in \mathcal{A}$ . Hence  $b = a$ , and thus  $\vdash_{\text{LiP}} a \mathbf{k} M' \rightarrow a \mathbf{k} \llbracket M' \rrbracket_b$  because  $\vdash_{\text{LiP}} a \mathbf{k} M' \rightarrow a \mathbf{k} \llbracket M' \rrbracket_a$ . Suppose that  $\vdash_{\text{LiP}} a \mathbf{k} M'$ . Hence  $\vdash_{\text{LiP}} a \mathbf{k} \llbracket M' \rrbracket_b$ , by *modus ponens*.
  - iii. *inductive step* for  $M := (M', M'')$ , for  $M', M'' \in \mathcal{M}$ . Suppose that  $\vdash_{\text{LiP}} a \mathbf{k} M'$  and  $\vdash_{\text{LiP}} a \mathbf{k} M''$ . Hence  $\vdash_{\text{LiP}} a \mathbf{k} M' \wedge a \mathbf{k} M''$ , by propositional logic. Now,  $\vdash_{\text{LiP}} (a \mathbf{k} M' \wedge a \mathbf{k} M'') \rightarrow a \mathbf{k} (M', M'')$ , and hence  $\vdash_{\text{LiP}} a \mathbf{k} (M', M'')$ , by *modus ponens*.
- (b) i.  $\vdash_{\text{LiP}} a \mathbf{k} (M, M')$  total knowledge  
 ii.  $\vdash_{\text{LiP}} a \mathbf{k} M \wedge a \mathbf{k} M'$  i, unpairing  
 iii.  $\vdash_{\text{LiP}} a \mathbf{k} M \leftrightarrow a \mathbf{k} M'$  ii, propositional logic.
- (c) Jointly from b and epistemic bitonicity by propositional logic.

## B.2 Proof of Theorem 2

1. (a)  $\vdash_{\text{LiP}} (M :_a^{\mathcal{C}} (\phi \rightarrow \phi')) \rightarrow ((M :_a^{\mathcal{C}} \phi) \rightarrow (M, M) :_a^{\mathcal{C}} \phi')$  GK  
 (b)  $\vdash_{\text{LiP}} ((M, M) :_a^{\mathcal{C}} \phi') \leftrightarrow M :_a^{\mathcal{C}} \phi'$  proof idempotency  
 (c)  $\vdash_{\text{LiP}} (M :_a^{\mathcal{C}} (\phi \rightarrow \phi')) \rightarrow ((M :_a^{\mathcal{C}} \phi) \rightarrow M :_a^{\mathcal{C}} \phi')$  a, b, PL.
2. (a)  $\{\phi \rightarrow \phi'\} \subseteq \text{LiP}$  hypothesis  
 (b)  $(\phi \rightarrow \phi') \in \text{LiP}$  a, definition  
 (c)  $\vdash_{\text{LiP}} \phi \rightarrow \phi'$  b, definition  
 (d)  $\vdash_{\text{LiP}} M :_a^{\mathcal{C}} (\phi \rightarrow \phi')$  c, necessitation  
 (e)  $\vdash_{\text{LiP}} (M :_a^{\mathcal{C}} (\phi \rightarrow \phi')) \rightarrow ((M :_a^{\mathcal{C}} \phi) \rightarrow M :_a^{\mathcal{C}} \phi')$  K  
 (f)  $\vdash_{\text{LiP}} (M :_a^{\mathcal{C}} \phi) \rightarrow M :_a^{\mathcal{C}} \phi'$  d, e, PL  
 (g)  $((M :_a^{\mathcal{C}} \phi) \rightarrow M :_a^{\mathcal{C}} \phi') \in \text{LiP}$  f, definition  
 (h)  $\{\phi \rightarrow \phi'\} \vdash_{\text{LiP}} (M :_a^{\mathcal{C}} \phi) \rightarrow M :_a^{\mathcal{C}} \phi'$  a-g, definition.
3. (a)  $\{\phi \leftrightarrow \phi'\} \subseteq \text{LiP}$  hypothesis  
 (b)  $(\phi \leftrightarrow \phi') \in \text{LiP}$  a, definition  
 (c)  $\vdash_{\text{LiP}} \phi \leftrightarrow \phi'$  b, definition  
 (d)  $\vdash_{\text{LiP}} \phi \rightarrow \phi'$  c, PL  
 (e)  $\vdash_{\text{LiP}} (M :_a^{\mathcal{C}} \phi) \rightarrow M :_a^{\mathcal{C}} \phi'$  d, regularity  
 (f)  $\vdash_{\text{LiP}} \phi' \rightarrow \phi$  c, PL  
 (g)  $\vdash_{\text{LiP}} (M :_a^{\mathcal{C}} \phi') \rightarrow M :_a^{\mathcal{C}} \phi$  f, regularity  
 (h)  $\vdash_{\text{LiP}} (M :_a^{\mathcal{C}} \phi) \leftrightarrow M :_a^{\mathcal{C}} \phi'$  e, g, PL  
 (i)  $((M :_a^{\mathcal{C}} \phi) \leftrightarrow M :_a^{\mathcal{C}} \phi') \in \text{LiP}$  h, definition  
 (j)  $\{\phi \leftrightarrow \phi'\} \vdash_{\text{LiP}} (M :_a^{\mathcal{C}} \phi) \leftrightarrow M :_a^{\mathcal{C}} \phi'$  a-i, definition.
4. By regularity, epistemic antitonicity, and the transitivity of ' $\rightarrow$ '.
5. By epistemic regularity and propositional logic.
6. (a)  $\vdash_{\text{LiP}} (M :_a^{\mathcal{C}} (\phi' \rightarrow (\phi \wedge \phi'))) \rightarrow ((M' :_a^{\mathcal{C}} \phi') \rightarrow (M, M') :_a^{\mathcal{C}} (\phi \wedge \phi'))$  GK  
 (b)  $\vdash_{\text{LiP}} \phi \rightarrow (\phi' \rightarrow (\phi \wedge \phi'))$  PT  
 (c)  $\vdash_{\text{LiP}} (M :_a^{\mathcal{C}} \phi) \rightarrow M :_a^{\mathcal{C}} (\phi' \rightarrow (\phi \wedge \phi'))$  b, regularity  
 (d)  $\vdash_{\text{LiP}} (M :_a^{\mathcal{C}} \phi) \rightarrow ((M' :_a^{\mathcal{C}} \phi') \rightarrow (M, M') :_a^{\mathcal{C}} (\phi \wedge \phi'))$  a, c, PL  
 (e)  $\vdash_{\text{LiP}} ((M :_a^{\mathcal{C}} \phi) \wedge M' :_a^{\mathcal{C}} \phi') \rightarrow (M, M') :_a^{\mathcal{C}} (\phi \wedge \phi')$  d, PL.
7. (a)  $\vdash_{\text{LiP}} ((M :_a^{\mathcal{C}} \phi) \wedge M :_a^{\mathcal{C}} \phi') \rightarrow (M, M) :_a^{\mathcal{C}} (\phi \wedge \phi')$  proof conjunctions  
 (b)  $\vdash_{\text{LiP}} ((M, M) :_a^{\mathcal{C}} (\phi \wedge \phi')) \leftrightarrow M :_a^{\mathcal{C}} (\phi \wedge \phi')$  proof idempotency  
 (c)  $\vdash_{\text{LiP}} ((M :_a^{\mathcal{C}} \phi) \wedge M :_a^{\mathcal{C}} \phi') \rightarrow M :_a^{\mathcal{C}} (\phi \wedge \phi')$  a, b, PL  
 (d)  $\vdash_{\text{LiP}} (\phi \wedge \phi') \rightarrow \phi$  PT

- (e)  $\vdash_{\text{LiP}} M :_a^{\mathcal{C}} (\phi \wedge \phi') \rightarrow M :_a^{\mathcal{C}} \phi$  d, regularity
- (f)  $\vdash_{\text{LiP}} (\phi \wedge \phi') \rightarrow \phi'$  PT
- (g)  $\vdash_{\text{LiP}} (M :_a^{\mathcal{C}} (\phi \wedge \phi')) \rightarrow M :_a^{\mathcal{C}} \phi'$  f, regularity
- (h)  $\vdash_{\text{LiP}} (M :_a^{\mathcal{C}} (\phi \wedge \phi')) \rightarrow ((M :_a^{\mathcal{C}} \phi) \wedge M :_a^{\mathcal{C}} \phi')$  e, g, PL
- (i)  $\vdash_{\text{LiP}} ((M :_a^{\mathcal{C}} \phi) \wedge M :_a^{\mathcal{C}} \phi') \leftrightarrow M :_a^{\mathcal{C}} (\phi \wedge \phi')$  c, h, PL.
  
- 8. (a)  $\vdash_{\text{LiP}} (M :_a^{\mathcal{C}} \phi) \rightarrow (M, M') :_a^{\mathcal{C}} \phi$  proof extension, right
- (b)  $\vdash_{\text{LiP}} \phi \rightarrow (\phi \vee \phi')$  PT
- (c)  $\vdash_{\text{LiP}} ((M, M') :_a^{\mathcal{C}} \phi) \rightarrow (M, M') :_a^{\mathcal{C}} (\phi \vee \phi')$  b, regularity
- (d)  $\vdash_{\text{LiP}} (M :_a^{\mathcal{C}} \phi) \rightarrow (M, M') :_a^{\mathcal{C}} (\phi \vee \phi')$  a, c, PL
- (e)  $\vdash_{\text{LiP}} (M' :_a^{\mathcal{C}} \phi') \rightarrow (M, M') :_a^{\mathcal{C}} \phi'$  proof extension, left
- (f)  $\vdash_{\text{LiP}} \phi' \rightarrow (\phi \vee \phi')$  PT
- (g)  $\vdash_{\text{LiP}} ((M, M') :_a^{\mathcal{C}} \phi') \rightarrow (M, M') :_a^{\mathcal{C}} (\phi \vee \phi')$  f, regularity
- (h)  $\vdash_{\text{LiP}} (M' :_a^{\mathcal{C}} \phi') \rightarrow (M, M') :_a^{\mathcal{C}} (\phi \vee \phi')$  e, g, PL
- (i)  $\vdash_{\text{LiP}} ((M :_a^{\mathcal{C}} \phi) \vee (M' :_a^{\mathcal{C}} \phi')) \rightarrow (M, M') :_a^{\mathcal{C}} (\phi \vee \phi')$  d, h, PL.
  
- 9. (a)  $\vdash_{\text{LiP}} ((M :_a^{\mathcal{C}} \phi) \vee M :_a^{\mathcal{C}} \phi') \rightarrow (M, M) :_a^{\mathcal{C}} (\phi \vee \phi')$  proof disjunctions
- (b)  $\vdash_{\text{LiP}} ((M, M) :_a^{\mathcal{C}} (\phi \vee \phi')) \leftrightarrow M :_a^{\mathcal{C}} (\phi \vee \phi')$  proof idempotency
- (c)  $\vdash_{\text{LiP}} ((M :_a^{\mathcal{C}} \phi) \vee M :_a^{\mathcal{C}} \phi') \rightarrow M :_a^{\mathcal{C}} (\phi \vee \phi')$  a, b, PL.
  
- 10. (a)  $\vdash_{\text{LiP}} a \mathbf{k} a$  knowledge of one's own name
- (b)  $\vdash_{\text{LiP}} \top$  a, definition
- (c)  $\vdash_{\text{LiP}} M :_a^{\mathcal{C}} \top$  b, necessitation.
  
- 11. (a)  $\vdash_{\text{LiP}} (a :_a^{\mathcal{C}} \phi) \rightarrow (a \mathbf{k} a \rightarrow \phi)$  epistemic truthfulness
- (b)  $\vdash_{\text{LiP}} a \mathbf{k} a$  knowledge of one's own name
- (c)  $\vdash_{\text{LiP}} a \mathbf{k} a \rightarrow ((a \mathbf{k} a \rightarrow \phi) \rightarrow \phi)$  PT
- (d)  $\vdash_{\text{LiP}} (a \mathbf{k} a \rightarrow \phi) \rightarrow \phi$  b, c, PL
- (e)  $\vdash_{\text{LiP}} (a :_a^{\mathcal{C}} \phi) \rightarrow \phi$  a, d, PL.
  
- 12. (a)  $\{\phi\} \vdash_{\text{LiP}} a :_a^{\mathcal{C}} \phi$  necessitation
- (b)  $\{a :_a^{\mathcal{C}} \phi\} \subseteq \text{LiP}$  hypothesis
- (c)  $(a :_a^{\mathcal{C}} \phi) \in \text{LiP}$  b, definition
- (d)  $\vdash_{\text{LiP}} a :_a^{\mathcal{C}} \phi$  c, definition
- (e)  $\vdash_{\text{LiP}} (a :_a^{\mathcal{C}} \phi) \rightarrow \phi$  self-truthfulness
- (f)  $\vdash_{\text{LiP}} \phi$  d, e, PL
- (g)  $\{a :_a^{\mathcal{C}} \phi\} \vdash_{\text{LiP}} \phi$  b-f, definition
- (h)  $\phi \Vdash_{\text{LiP}} a :_a^{\mathcal{C}} \phi$  a, g, definition.

13. (a)  $\vdash_{\text{LiP}} (M :_a^{\mathcal{C}} \perp) \rightarrow (a \text{ k } M \rightarrow \perp)$  epistemic truthfulness  
 (b)  $\vdash_{\text{LiP}} ((M :_a^{\mathcal{C}} \perp) \wedge a \text{ k } M) \rightarrow \perp$  a, PL  
 (c)  $\vdash_{\text{LiP}} (((M :_a^{\mathcal{C}} \perp) \wedge a \text{ k } M) \rightarrow \perp) \rightarrow \neg((M :_a^{\mathcal{C}} \perp) \wedge a \text{ k } M)$  PT  
 (d)  $\vdash_{\text{LiP}} \neg((M :_a^{\mathcal{C}} \perp) \wedge a \text{ k } M)$  b, c, PL  
 (e)  $\vdash_{\text{LiP}} a \text{ k } M \rightarrow \neg(M :_a^{\mathcal{C}} \perp)$  d, PL.
14. By the preceding law and knowledge of one's own name string.
15. (a)  $\vdash_{\text{LiP}} a \text{ k } M \rightarrow \neg(M :_a^{\mathcal{C}} \perp)$  nothing known can prove falsehood  
 (b)  $\vdash_{\text{LiP}} (\phi \wedge \neg\phi) \leftrightarrow \perp$  PT  
 (c)  $\vdash_{\text{LiP}} a \text{ k } M \rightarrow \neg(M :_a^{\mathcal{C}} (\phi \wedge \neg\phi))$  a, b, PL  
 (d)  $\vdash_{\text{LiP}} ((M :_a^{\mathcal{C}} \phi) \wedge M :_a^{\mathcal{C}} \neg\phi) \leftrightarrow M :_a^{\mathcal{C}} (\phi \wedge \neg\phi)$  proof conjunctions *bis*  
 (e)  $\vdash_{\text{LiP}} a \text{ k } M \rightarrow \neg((M :_a^{\mathcal{C}} \phi) \wedge M :_a^{\mathcal{C}} \neg\phi)$  c, d, PL  
 (f)  $\vdash_{\text{LiP}} a \text{ k } M \rightarrow ((M :_a^{\mathcal{C}} \phi) \rightarrow \neg(M :_a^{\mathcal{C}} \neg\phi))$  e, PL  
 (g)  $\vdash_{\text{LiP}} a \text{ k } M \rightarrow ((M :_a^{\mathcal{C}} \phi) \rightarrow M \diamond_a^{\mathcal{C}} \phi)$  f, definition.
16. By the preceding law and knowledge of one's own name string.
17. (a)  $\vdash_{\text{LiP}} (M :_b^{\mathcal{C}} \top) \rightarrow \bigwedge_{a \in \mathcal{C} \cup \{b\}} (\llbracket M \rrbracket_b :_a^{\mathcal{C} \cup \{b\}} (b \text{ k } M \wedge M :_b^{\mathcal{C}} \top))$  peer review  
 (b)  $\vdash_{\text{LiP}} M :_b^{\mathcal{C}} \top$  anything can prove truth  
 (c)  $\vdash_{\text{LiP}} \bigwedge_{a \in \mathcal{C} \cup \{b\}} (\llbracket M \rrbracket_b :_a^{\mathcal{C} \cup \{b\}} (b \text{ k } M \wedge M :_b^{\mathcal{C}} \top))$  a, b, PL  
 (d)  $\vdash_{\text{LiP}} \llbracket M \rrbracket_b :_a^{\mathcal{C} \cup \{b\}} (b \text{ k } M \wedge M :_b^{\mathcal{C}} \top)$   $b \in \mathcal{C} \cup \{b\}$ , c, PL  
 (e)  $\vdash_{\text{LiP}} (\llbracket M \rrbracket_b :_a^{\mathcal{C} \cup \{b\}} (b \text{ k } M \wedge M :_b^{\mathcal{C}} \top)) \rightarrow \llbracket M \rrbracket_b :_a^{\mathcal{C} \cup \{b\}} b \text{ k } M$  proof conj. *bis*  
 (f)  $\vdash_{\text{LiP}} \llbracket M \rrbracket_b :_a^{\mathcal{C} \cup \{b\}} b \text{ k } M$  d, e, PL.
18. (a)  $\vdash_{\text{LiP}} \llbracket M \rrbracket_a :_a^{\emptyset \cup \{a\}} a \text{ k } M$  authentic knowledge  
 (b)  $\vdash_{\text{LiP}} (\llbracket M \rrbracket_a :_a^{\emptyset \cup \{a\}} a \text{ k } M) \rightarrow \llbracket M \rrbracket_a :_a^{\emptyset} a \text{ k } M$  group decomposition  
 (c)  $\vdash_{\text{LiP}} \llbracket M \rrbracket_a :_a^{\emptyset} a \text{ k } M$  a, b, PL  
 (d)  $\vdash_{\text{LiP}} (\llbracket M \rrbracket_a :_a^{\emptyset} a \text{ k } M) \rightarrow M :_a^{\emptyset} a \text{ k } M$  self-signing elimination  
 (e)  $\vdash_{\text{LiP}} M :_a^{\emptyset} a \text{ k } M$  c, d, PL.
19. (a)  $\vdash_{\text{LiP}} (M :_a^{\mathcal{C}} \phi) \rightarrow \bigwedge_{b \in \mathcal{C} \cup \{a\}} (\llbracket M \rrbracket_a :_b^{\mathcal{C} \cup \{a\}} (a \text{ k } M \wedge M :_a^{\mathcal{C}} \phi))$  peer review  
 (b)  $b \in \mathcal{C} \cup \{a\}$  hypothesis  
 (c)  $\vdash_{\text{LiP}} (M :_a^{\mathcal{C}} \phi) \rightarrow (a \text{ k } M \rightarrow \phi)$  epistemic truthfulness  
 (d)  $\vdash_{\text{LiP}} (a \text{ k } M \wedge M :_a^{\mathcal{C}} \phi) \rightarrow \phi$  c, PL  
 (e)  $\vdash_{\text{LiP}} (\llbracket M \rrbracket_a :_b^{\mathcal{C} \cup \{a\}} (a \text{ k } M \wedge M :_a^{\mathcal{C}} \phi)) \rightarrow \llbracket M \rrbracket_a :_b^{\mathcal{C} \cup \{a\}} \phi$  d, regularity  
 (f)  $\vdash_{\text{LiP}} \bigwedge_{b \in \mathcal{C} \cup \{a\}} (\llbracket M \rrbracket_a :_b^{\mathcal{C} \cup \{a\}} (a \text{ k } M \wedge M :_a^{\mathcal{C}} \phi)) \rightarrow \llbracket M \rrbracket_a :_b^{\mathcal{C} \cup \{a\}} \phi$  b-e, PL



- (g)  $\vdash_{\text{LiP}} (M :_a^{\mathcal{C}} \phi) \rightarrow \bigwedge_{b \in \mathcal{C} \cup \{a\}} (\llbracket M \rrbracket_a :_b^{\mathcal{C} \cup \{a\}} \phi)$  a, f, PL.
20. (a)  $\vdash_{\text{LiP}} (M :_a^{\mathcal{C} \cup \mathcal{C}'} \phi) \rightarrow (M :_a^{\mathcal{C}} \phi)$  group decomposition  
 (b)  $\vdash_{\text{LiP}} (M :_a^{\mathcal{C} \cup \mathcal{C}'} \phi) \leftrightarrow (M :_a^{\mathcal{C}' \cup \mathcal{C}} \phi)$   $\mathcal{C} \cup \mathcal{C}' = \mathcal{C}' \cup \mathcal{C}$   
 (c)  $\vdash_{\text{LiP}} (M :_a^{\mathcal{C}' \cup \mathcal{C}} \phi) \rightarrow (M :_a^{\mathcal{C}'} \phi)$  group decomposition  
 (d)  $\vdash_{\text{LiP}} (M :_a^{\mathcal{C} \cup \mathcal{C}'} \phi) \rightarrow (M :_a^{\mathcal{C}'} \phi)$  b, c, PL  
 (e)  $\vdash_{\text{LiP}} (M :_a^{\mathcal{C} \cup \mathcal{C}'} \phi) \rightarrow ((M :_a^{\mathcal{C}} \phi) \wedge M :_a^{\mathcal{C}'} \phi)$  a, d, PL.
21. (a)  $\vdash_{\text{LiP}} (M :_a^{\mathcal{C} \cup \{a\}} \phi) \rightarrow (M :_a^{\mathcal{C}} \phi)$  group decomposition  
 (b)  $\vdash_{\text{LiP}} (M :_a^{\mathcal{C}} \phi) \rightarrow \bigwedge_{b \in \mathcal{C} \cup \{a\}} (\llbracket M \rrbracket_a :_b^{\mathcal{C} \cup \{a\}} \phi)$  simple peer review  
 (c)  $\vdash_{\text{LiP}} (M :_a^{\mathcal{C}} \phi) \rightarrow \llbracket M \rrbracket_a :_a^{\mathcal{C} \cup \{a\}} \phi$   $a \in \mathcal{C} \cup \{a\}$ , b, PL  
 (d)  $\vdash_{\text{LiP}} (\llbracket M \rrbracket_a :_a^{\mathcal{C} \cup \{a\}} \phi) \leftrightarrow M :_a^{\mathcal{C} \cup \{a\}} \phi$  self-signing idempotency  
 (e)  $\vdash_{\text{LiP}} (M :_a^{\mathcal{C}} \phi) \rightarrow M :_a^{\mathcal{C} \cup \{a\}} \phi$  c, d, PL  
 (f)  $\vdash_{\text{LiP}} (M :_a^{\mathcal{C} \cup \{a\}} \phi) \leftrightarrow M :_a^{\mathcal{C}} \phi$  a, e, PL.
22. (a)  $\vdash_{\text{LiP}} a \mathbf{k} M \rightarrow ((M :_a^{\mathcal{C}} \phi) \rightarrow \phi)$  epistemic truthfulness, PL  
 (b)  $\vdash_{\text{LiP}} (\llbracket M \rrbracket_a :_a^{\mathcal{C}} a \mathbf{k} M) \rightarrow \llbracket M \rrbracket_a :_a^{\mathcal{C}} ((M :_a^{\mathcal{C}} \phi) \rightarrow \phi)$  a, regularity  
 (c)  $\vdash_{\text{LiP}} \llbracket M \rrbracket_a :_a^{\mathcal{C} \cup \{a\}} a \mathbf{k} M$  authentic knowledge  
 (d)  $\vdash_{\text{LiP}} (\llbracket M \rrbracket_a :_a^{\mathcal{C} \cup \{a\}} a \mathbf{k} M) \leftrightarrow \llbracket M \rrbracket_a :_a^{\mathcal{C}} a \mathbf{k} M$  self-neutral group element  
 (e)  $\vdash_{\text{LiP}} \llbracket M \rrbracket_a :_a^{\mathcal{C}} a \mathbf{k} M$  c, d, PL  
 (f)  $\vdash_{\text{LiP}} \llbracket M \rrbracket_a :_a^{\mathcal{C}} ((M :_a^{\mathcal{C}} \phi) \rightarrow \phi)$  b, e, PL  
 (g)  $\vdash_{\text{LiP}} (\llbracket M \rrbracket_a :_a^{\mathcal{C}} ((M :_a^{\mathcal{C}} \phi) \rightarrow \phi)) \leftrightarrow M :_a^{\mathcal{C}} ((M :_a^{\mathcal{C}} \phi) \rightarrow \phi)$  self-signing idempotency  
 (h)  $\vdash_{\text{LiP}} M :_a^{\mathcal{C}} ((M :_a^{\mathcal{C}} \phi) \rightarrow \phi)$  f, g, PL.
23. By instantiating the previous law with ‘ $\perp$ ’, regularity *bis*, and propositional logic.
24. (a)  $\vdash_{\text{LiP}} (M :_a^{\mathcal{C}} \phi) \rightarrow \bigwedge_{b \in \mathcal{C} \cup \{a\}} (\llbracket M \rrbracket_a :_b^{\mathcal{C} \cup \{a\}} (a \mathbf{k} M \wedge M :_a^{\mathcal{C}} \phi))$  peer review  
 (b)  $b \in \mathcal{C} \cup \{a\}$  hypothesis  
 (c)  $\vdash_{\text{LiP}} (M :_a^{\mathcal{C}} \phi) \rightarrow \bigwedge_{b \in \mathcal{C} \cup \{a\}} (\llbracket M \rrbracket_a :_b^{\mathcal{C} \cup \{a\}} \phi)$  simple peer review  
 (d)  $\vdash_{\text{LiP}} (a \mathbf{k} M \wedge M :_a^{\mathcal{C}} \phi) \rightarrow \bigwedge_{b \in \mathcal{C} \cup \{a\}} (\llbracket M \rrbracket_a :_b^{\mathcal{C} \cup \{a\}} \phi)$  c, PL  
 (e)  $\vdash_{\text{LiP}} (\llbracket M \rrbracket_a :_b^{\mathcal{C} \cup \{a\}} (a \mathbf{k} M \wedge M :_a^{\mathcal{C}} \phi)) \rightarrow \llbracket M \rrbracket_a :_b^{\mathcal{C} \cup \{a\}} (\bigwedge_{b \in \mathcal{C} \cup \{a\}} (\llbracket M \rrbracket_a :_b^{\mathcal{C} \cup \{a\}} \phi))$  d, regularity  
 (f)  $\vdash_{\text{LiP}} \bigwedge_{b \in \mathcal{C} \cup \{a\}} ((\llbracket M \rrbracket_a :_b^{\mathcal{C} \cup \{a\}} (a \mathbf{k} M \wedge M :_a^{\mathcal{C}} \phi)) \rightarrow \llbracket M \rrbracket_a :_b^{\mathcal{C} \cup \{a\}} (\bigwedge_{b \in \mathcal{C} \cup \{a\}} (\llbracket M \rrbracket_a :_b^{\mathcal{C} \cup \{a\}} \phi)))$  b-e, PL

- (g)  $\vdash_{\text{LiP}} (M :_a^{\mathcal{C}} \phi) \rightarrow \bigwedge_{b \in \mathcal{C} \cup \{a\}} (\llbracket M \rrbracket_a :_b^{\mathcal{C} \cup \{a\}} (\bigwedge_{b \in \mathcal{C} \cup \{a\}} (\llbracket M \rrbracket_a :_b^{\mathcal{C} \cup \{a\}} \phi)))$  a, f, PL
- (h)  $\vdash_{\text{LiP}} (M :_a^{\mathcal{C}} \phi) \rightarrow \llbracket M \rrbracket_a :_a^{\mathcal{C} \cup \{a\}} (\bigwedge_{b \in \mathcal{C} \cup \{a\}} (\llbracket M \rrbracket_a :_b^{\mathcal{C} \cup \{a\}} \phi))$   $a \in \mathcal{C} \cup \{a\}$ , g, PL
- (i)  $\vdash_{\text{LiP}} \llbracket M \rrbracket_a :_a^{\mathcal{C} \cup \{a\}} (\bigwedge_{b \in \mathcal{C} \cup \{a\}} (\llbracket M \rrbracket_a :_b^{\mathcal{C} \cup \{a\}} \phi)) \leftrightarrow \llbracket M \rrbracket_a :_a^{\mathcal{C}} (\bigwedge_{b \in \mathcal{C} \cup \{a\}} (\llbracket M \rrbracket_a :_b^{\mathcal{C} \cup \{a\}} \phi))$  self-neutral group element
- (j)  $\vdash_{\text{LiP}} (M :_a^{\mathcal{C}} \phi) \rightarrow \llbracket M \rrbracket_a :_a^{\mathcal{C}} (\bigwedge_{b \in \mathcal{C} \cup \{a\}} (\llbracket M \rrbracket_a :_b^{\mathcal{C} \cup \{a\}} \phi))$  h, i, PL
- (k)  $\vdash_{\text{LiP}} (\llbracket M \rrbracket_a :_a^{\mathcal{C}} (\bigwedge_{b \in \mathcal{C} \cup \{a\}} (\llbracket M \rrbracket_a :_b^{\mathcal{C} \cup \{a\}} \phi))) \leftrightarrow M :_a^{\mathcal{C}} (\bigwedge_{b \in \mathcal{C} \cup \{a\}} (\llbracket M \rrbracket_a :_b^{\mathcal{C} \cup \{a\}} \phi))$  self-signing idempotency
- (l)  $\vdash_{\text{LiP}} (M :_a^{\mathcal{C}} \phi) \rightarrow M :_a^{\mathcal{C}} (\bigwedge_{b \in \mathcal{C} \cup \{a\}} (\llbracket M \rrbracket_a :_b^{\mathcal{C} \cup \{a\}} \phi))$  j, k, PL.
25. (a)  $\vdash_{\text{LiP}} (M :_a^{\mathcal{C}} \phi) \rightarrow M :_a^{\mathcal{C}} (\llbracket M \rrbracket_a :_a^{\mathcal{C} \cup \{a\}} \phi)$  simple peer review *bis*
- (b)  $\vdash_{\text{LiP}} (\llbracket M \rrbracket_a :_a^{\mathcal{C} \cup \{a\}} \phi) \leftrightarrow \llbracket M \rrbracket_a :_a^{\mathcal{C}} \phi$  self-neutral group element
- (c)  $\vdash_{\text{LiP}} (M :_a^{\mathcal{C}} (\llbracket M \rrbracket_a :_a^{\mathcal{C} \cup \{a\}} \phi)) \leftrightarrow M :_a^{\mathcal{C}} (\llbracket M \rrbracket_a :_a^{\mathcal{C}} \phi)$  b, regularity *bis*
- (d)  $\vdash_{\text{LiP}} (M :_a^{\mathcal{C}} \phi) \rightarrow M :_a^{\mathcal{C}} (\llbracket M \rrbracket_a :_a^{\mathcal{C}} \phi)$  a, c, PL
- (e)  $\vdash_{\text{LiP}} (\llbracket M \rrbracket_a :_a^{\mathcal{C}} \phi) \leftrightarrow M :_a^{\mathcal{C}} \phi$  self-signing idempotency
- (f)  $\vdash_{\text{LiP}} (M :_a^{\mathcal{C}} (\llbracket M \rrbracket_a :_a^{\mathcal{C}} \phi)) \leftrightarrow M :_a^{\mathcal{C}} (M :_a^{\mathcal{C}} \phi)$  e, regularity *bis*
- (g)  $\vdash_{\text{LiP}} (M :_a^{\mathcal{C}} \phi) \rightarrow M :_a^{\mathcal{C}} (M :_a^{\mathcal{C}} \phi)$  d, f, PL
- (h)  $\vdash_{\text{LiP}} M :_a^{\mathcal{C}} ((M :_a^{\mathcal{C}} \phi) \rightarrow \phi)$  self-proof of proof consistency
- (i)  $\vdash_{\text{LiP}} (M :_a^{\mathcal{C}} ((M :_a^{\mathcal{C}} \phi) \rightarrow \phi)) \rightarrow ((M :_a^{\mathcal{C}} (M :_a^{\mathcal{C}} \phi)) \rightarrow M :_a^{\mathcal{C}} \phi)$  K
- (j)  $\vdash_{\text{LiP}} (M :_a^{\mathcal{C}} (M :_a^{\mathcal{C}} \phi)) \rightarrow M :_a^{\mathcal{C}} \phi$  h, i, PL
- (k)  $\vdash_{\text{LiP}} (M :_a^{\mathcal{C}} (M :_a^{\mathcal{C}} \phi)) \leftrightarrow M :_a^{\mathcal{C}} \phi$  g, j, PL.
26. Jointly from the law of total knowledge, and the law that nothing known can prove falsehood, epistemic truthfulness, and proof consistency, respectively.